

SAP Database Guide: Oracle (BC-DB-ORA-DBA)

SAP Web Application Server Release 6.20



Copyright

© Copyright 2002 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®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corp. in USA and/or other countries.

ORACLE® is a registered trademark of ORACLE Corporation.

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

Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

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, 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.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Documentation in the SAP Service Marketplace






You can find this documentation at the following address:

<http://service.sap.com/instguides>

Typographic Conventions

Type Style	Represents
<i>Example Text</i>	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, titles of graphics and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, source code as well as names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.
EXAMPLE TEXT	Keys on the keyboard, for example, function keys (such as F2) or the ENTER key

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

SAP Database Guide: Oracle (BC-DB-ORA-DBA)	15
1 New Features in SAP Release 6.20	16
2 New Features in SAP Release 6.10	18
3 New Features in SAP R/3 Release 4.6	19
4 New Features in SAP R/3 Release 4.5	20
4.1 Release 4.5: SAPDBA	20
4.2 Release 4.5: BRBACKUP, BRARCHIVE, and BRRESTORE	20
5 New Features in SAP R/3 Release 4.0	22
Getting Started with Oracle and the SAP System	23
6 Computing Center Management System	24
7 Specification of SAP Tables in the ABAP Dictionary	25
8 Oracle Database Storage Parameters in the ABAP Dictionary	26
9 Support with SAP Service Marketplace	28
10 GoingLive and EarlyWatch	29
11 Oracle Databases on Raw Devices	33
11.1 Raw Devices and SAPDBA	33
11.2 Raw Devices with BRBACKUP and BRRESTORE	34
12 Limitations of the Oracle Database System	36
13 Database System Configuration	38
13.1 Changing Database User Passwords	39
13.2 Initializing SAP Connect	40
13.3 Setting Archive and Database Mode with SAPDBA	41
13.4 Setting Up Archiving	43
13.5 Mirroring the Control File	45
13.6 Mirroring Online Redo Log Files	45
13.7 Organizing Disk Storage	46
13.8 Database Parameters	47
13.9 SAP Naming Conventions for Tablespaces and Data Files	49
13.10 Environment Variables (UNIX)	51
13.11 Environment Variables (Windows NT)	53
13.12 Directory Structure (UNIX)	55
13.13 Directory Structure (Windows NT)	60
Approach to Oracle DBA	66
14 Database Mode	67
15 Database System Check	69
16 Space Management	70
17 Update Statistics	71
Backup, Restore, and Recovery	73
18 Backup Overview	75
18.1 Why Back Up the Database?	76
18.2 What Needs Backing Up?	77
18.3 Database Backup Types	80
18.3.1 Online and Offline Backup	81
18.3.2 Consistent Online Backup	82
18.3.3 Complete Backup	83
18.3.4 Incremental Backup	83
18.4 Backup Cycles	84

18.5 Backup Approach with Daily Complete Backups	85
18.6 Backup Approach for Very Large Database with Partial Backups	87
18.7 Backup Approach with Retention Period Zero	88
19 Backup Media	90
19.1 Volume Management	91
19.1.1 Volume Initialization	91
19.1.2 Volume Label Check	93
19.1.3 Volume Expiration Period	95
19.1.4 Used Volumes	96
19.1.5 Scratch Volume	98
19.1.6 Selecting Volumes Manually	98
19.1.7 Selecting Volumes with External Tools	99
19.1.8 Selecting Volumes Automatically	100
19.2 Hardware Compression	101
19.3 Software Compression	102
20 Backup Methods	105
20.1 Backup to Multiple Disks	105
20.2 Backup to a Remote Disk	105
20.3 Backup to a Remote Tape Device	108
20.4 Two-Phase Backup	109
20.5 Structure-Retaining Database Copy	111
20.6 Parallel Backup	113
20.7 Unattended Backup	114
20.8 BRBACKUP and BRARCHIVE Backups in One Run	116
20.9 Backup Verify	118
21 Approach to Restore and Recovery	121
21.1 Disaster Recovery	123
21.2 Partial Restore and Complete Recovery	127
21.3 Full Restore and Point-In-Time Recovery	130
21.4 Database Reset	133
22 Oracle Recovery Manager	137
22.1 RMAN Backup Strategies	138
22.2 RMAN Incremental Backups After Structural Changes	140
22.3 RMAN Restore of Incremental Backups	141
22.4 RMAN Backup with the SAP Backup Library	142
22.5 RMAN Backup with an External Backup Library	144
22.6 RMAN Incremental Backups Without a Backup Library	146
22.7 RMAN Backup of the Offline Redo Log File	149
22.8 RMAN Tape Layout	149
22.9 RMAN Backup Verify	150
22.10 RMAN Save-Set Grouping	151
23 Advanced Backup and Recovery	153
23.1 Backup of Large Oracle Databases	153
23.1.1 Backup Devices for Large Databases	154
23.1.2 Backup of Large Databases to Tape with BRBACKUP	154
23.1.3 External Backup Programs for Large Databases	155
23.1.4 Parallel Backup of Large Databases to Disk with BRBACKUP	156
23.1.5 Optimization of File Distribution	156
23.1.6 Optimization with a Logical Volume Manager	158
23.1.7 Partial Backups	159
23.1.8 Tablespace Backups	160
23.1.9 Backup Example for a Large Database	161

23.1.10 Speeding Up the Backup	161
23.2 Standby Database	162
23.2.1 Standby Database Configuration	163
23.2.2 Standby Database: BRARCHIVE Backup of Offline Redo Log Files	165
23.2.3 Standby Database: BRBACKUP Backup of Database Files	167
23.2.4 Standby Database: Restore and Recovery	168
23.2.5 Standby Database: Remote Database Connect Requirements	169
23.3 Split-Mirror Disk Backup	170
23.3.1 Split-Mirror Online Backup	172
23.3.2 Split-Mirror Offline Backup	173
23.3.3 Split-Mirror Backup: Software Configuration	174
23.4 Backup with Automatic Tape Changers	176
23.4.1 Mount and Dismount Commands	177
23.4.2 Autoloader Backup Example	178
23.5 Veritas Quick I/O Feature	179
23.6 External Backup Programs	180
Tools for Oracle DBA	183
24 SAPDBA for Oracle	184
24.1 Getting Started with SAPDBA	185
24.1.1 SAPDBA Configuration	185
Configuring the SAPDBA Display Length	186
Configuring the UNIX Command at	186
Setting the Log Checks Option	187
Users and Roles	187
UNIX User Configurations	189
Effects of Autoextend and Resize on SAPDBA	192
24.1.2 Starting SAPDBA	193
Checking SAPDBA Release Information	195
SAPDBA Main Menu	196
24.1.3 Show/Cleanup to Manage Logs and Profiles with SAPDBA	197
Initialization Profile init<DBSID>.dba with SAPDBA	198
backup_util_name	198
blkdev_dir	199
check_controlfile_mirror	199
check_initial_settings	199
check_redolog_mirror	200
exe_dir	200
exireo_dumpdir	200
exireo_exp_imp_buf	200
expir_period_brarchive	201
exp_imp_degree	201
file_or_raw	201
index_degree	201
max_file_size	201
message_buf	202
orapwd_always_used	202
parallel_instances	202
rawdev_dir	202
recovery_degree	203

sqldba_path	203
trace_level	203
tspadd_tsname	203
Sample Initialization Profile	204
SAPDBA Log Files and Their Contents	205
Structure Log	206
Central Log reorg<DBSID>.log	207
Return Codes in the Central Log reorg<DBSID>.log	209
Logs for Parallel Export and Import Processes	210
Logging Information in CCMS	211
Deleting Log, Directory, Audit, and Trace Files	212
24.1.4 SAPDBA Scripts and Files	214
Starting SQL Scripts Manually with SQLPLUS	216
Starting Individual SQL Scripts with SAPDBA	217
Starting Shell Scripts for Oracle Export and Import	217
24.2 Startup and Shutdown of the Oracle Database with SAPDBA	217
24.2.1 Checking the Status of the Oracle Database System	218
24.2.2 Starting Up the Oracle Database System	218
24.2.3 Shutting Down the Oracle Database System	219
24.2.4 Switching to Restricted Session Mode	221
24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA	222
24.3.1 Updating Statistics using Integrated Approach with SAPDBA	223
24.3.2 Updating Statistics using Optimizer Control with SAPDBA	226
24.3.3 Update Statistics: Internal Rules with SAPDBA	227
24.4 Tablespace Administration with SAPDBA	229
24.4.1 Check Tablespaces	230
Checking Freespace for Tablespaces	231
Checking Freespace for Objects in Tablespace	232
Preventing Tablespace Overflow	233
24.4.2 Extending a Tablespace	233
24.4.3 Creating or Dropping a Tablespace	235
24.4.4 Specifying the Path for the New Data File	238
24.4.5 Specifying the Size for the New Data File	239
24.4.6 Checks for Tablespace Extension and Creation	240
24.4.7 Backing Up After Tablespace Operations	241
24.4.8 Displaying File and Raw Device Information for a Tablespace	242
24.5 Reorganization with SAPDBA	242
24.5.1 Preparations for Reorganization	244
Checking Statistics for a Reorganization	245
Checking Extents and Fragmentation	246
Deciding if a Reorganization is Necessary	247
Performance Improvements for Reorganization	248
Changing the Initialization Profile to Speed Up Reorganization	248
Performing Processes in Parallel to Speed Up Reorganization	248
Reorganizing in NOARCHIVELOG Mode	249
Choosing a Reorganization Type	251
Backing Up the Database for Reorganization	252
24.5.2 Change Storage Parameters with SAPDBA	253
Viewing and Changing Storage Parameters	254

Storage Parameters	256
Default Value for NEXT	257
24.5.3 Reorganization	259
Reorganizing a Single Table or Index	260
Reorganizing a List of Tables and Indexes	263
Reorganizing a Tablespace	267
Reorganization of Special Tablespaces	270
Reorganizing a Tablespace and Data Files	271
Moving or Renaming the Data Files of a Tablespace	274
Resizing the Data Files of a Tablespace	275
Restarting a Reorganization	276
Restart File for Reorganizations	279
Freespace Check for a Reorganization	280
Background Reorganization	283
Reorganization Methods	284
Storage Parameters for Reorganization and Export/Import	284
24.6 Export and Import of Database Objects with SAPDBA	285
24.6.1 Preparations for Export and Import	287
Types of Export and Import	288
Parallel Export and Import	289
24.6.2 Export with SAPDBA	290
Exporting Tables and Indexes Including Data	291
Exporting Structures of Tables and Indexes	293
Exporting Constraints, Synonyms, Sequences, Views, and Grants	295
Storage Parameters for Reorganization and Export/Import	296
24.6.3 Import with SAPDBA	297
Importing Tables and Indexes Including Data	298
Importing Structures	299
24.6.4 Restarting an Export or Import	300
Restart File for Export or Import	302
24.7 Database Backup with SAPDBA	303
24.7.1 Backing Up the Database with SAPDBA	304
Setting Parameters for SAPDBA Database Backup	306
Setting Special Options for SAPDBA Database Backup	311
24.8 Backup of the Offline Redo Log Files with SAPDBA	313
24.8.1 Backing Up the Offline Redo Log Files with SAPDBA	314
Setting Parameters for SAPDBA Backup of Offline Redo Logs	315
Setting Special Options for SAPDBA Backup of Offline Redo Logs	318
24.8.2 Grouping Offline Redo Log Files	320
24.9 Database Restore and Recovery with SAPDBA	321
24.9.1 Partial Restore and Complete Recovery with SAPDBA	323
Checking the Database	324
Finding the Backup Files	326
Restoring the Backup Files	327
Finding Incremental Backups	328
Restoring Incremental Backups	330
Finding the Offline Redo Log Files	331
Restoring the Offline Redo Log Files	332
Recovering the Database	333

Recovering the Database Automatically	334
Changing the Restore and Recovery Options	334
24.9.2 Performing a Full Restore and Point-In-Time Recovery with SAPDBA	335
24.9.3 Resetting the Database	340
24.9.4 Restoring One Tablespace	341
24.9.5 Restoring Individual Files	341
24.10 SAPDBA Command Options	344
24.10.1 sapdba -alter_user	345
24.10.2 sapdba -analyze	345
24.10.3 sapdba -check	347
24.10.4 sapdba -check_db_open	348
24.10.5 sapdba -checkopt	348
24.10.6 sapdba -cleanup	350
24.10.7 sapdba -delete	350
24.10.8 sapdba -export	350
24.10.9 sapdba -fill_tables	351
24.10.10 sapdba -init_sap_connect	351
24.10.11 sapdba -next <tablespace(s)>	351
24.10.12 sapdba -r	352
24.10.13 sapdba -shutdown(_abort)	353
24.10.14 sapdba -startup(_dba)	353
24.10.15 sapdba -statistics	353
24.10.16 sapdba -version	354
24.10.17 Return Codes for SAPDBA Command Options	355
25 BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT	356
25.1 Common Features of BRBACKUP and BRARCHIVE	357
25.1.1 Supported Backup Media	358
25.1.2 Effects of the Command Options	359
25.1.3 cpio Continuation Tape	359
25.1.4 cpio Error	360
25.1.5 Canceling a Backup	360
25.2 BRBACKUP	360
25.2.1 Backing up Database Files	361
25.2.2 Backing Up Non-Database Files and Directories	361
25.2.3 Completion of BRBACKUP Backups	362
25.2.4 Hardware Compression for BRBACKUP	363
25.2.5 Command Options for BRBACKUP	363
-a -archive	364
-b -backup	365
-bd -backup_delete	365
-c -confirm	366
-db -delete_backup	367
-d -device	367
-e -execute	368
-f -fillup	369
-h -help	369
-i -initialize	369
-k -compress	370
-l -language	370

-m -mode	370
-n -number	372
-o -output	372
-p -profile	372
-q -query	373
-r -parfile	373
-s -saveaset	374
-t -type	374
-u -user	375
-v -volume	375
-w -verify	376
-V -VERSION	377
25.2.6 BRBACKUP Logs	377
Names of the BRBACKUP Detail Logs	377
BRBACKUP Detail Log	378
BRBACKUP Summary Log	379
25.2.7 Profile Parameters and BRBACKUP Command Options	380
25.3 BRARCHIVE	383
25.3.1 Hardware Compression for BRARCHIVE	383
25.3.2 Logging for BRARCHIVE	384
25.3.3 Command Options for BRARCHIVE	384
-a -archive	385
-b -backup	386
-c -confirm	386
-d -device	387
-f -fill	387
-h -help	388
-i -initialize	388
-k -compress	388
-l -language	388
-m -modify	389
-n -number	389
-o -output	389
-p -profile	389
-q -query	389
-r -parfile	390
-s -sc -ds -dc -sd -scd -ss -ssd -cs -cds	390
-u -user	391
-v -volume	391
-w -verify	391
-V -VERSION	392
25.3.4 BRARCHIVE Logs	392
Names of the BRARCHIVE Detail Logs	392
BRARCHIVE Detail Log	393
BRARCHIVE Summary Log	394
25.4 BRRESTORE	395
25.4.1 Restoring Files	396
25.4.2 Completion of BRRESTORE Runs	396
25.4.3 Examples of BRRESTORE Runs	397

25.4.4 Command Options for BRRESTORE	398
-a -archive -a1 -archive1	399
-a2 -archive2	400
-b -backup b1 backup1	400
-b2 -backup2	400
-c -confirm	401
-d -device	401
-e -execute	402
-f -fillup	402
-h -help	402
-k -compress	402
-l -language	403
-m -mode	403
-n -number	404
-o -output	404
-p -profile	405
-q -query	405
-r -parfile	405
-w -verify	405
-V -VERSION	406
25.4.5 BRRESTORE Logs	406
Names of the BRRESTORE Detail Logs	406
BRRESTORE Detail Log	406
BRRESTORE Summary Log	407
25.5 BRCONNECT	407
25.5.1 Database System Check with BRCONNECT	409
BRCONNECT Default Conditions for Database Administration	411
BRCONNECT Default Conditions for Database Operations	416
25.5.2 Adapt Next Extents with BRCONNECT	419
Internal Rules for Determining Next Extent Size	419
25.5.3 Update Statistics with BRCONNECT	420
-force with Update Statistics	422
Deletion of Damaging Statistics	423
Verification of Table and Index Structure	423
Internal Rules for Update Statistics	424
Update Statistics for InfoCube Tables	425
25.5.4 Clean Up Old Logs and Traces with BRCONNECT	427
25.5.5 Additional BRCONNECT Functions	428
25.5.6 Command Options for BRCONNECT	429
-c -confirm	430
-f -function	430
-f check	431
-f chpass	432
-f cleanup	432
-f crsyn	434
-f dbshut	434
-f next	435
-f stats	436
-h -help	441

-l -language	441
-o -output	441
-p -profile	441
-q -query	442
-s -sapsid	442
-u -user	442
-V -VERSION	443
25.5.7 BRCONNECT Logs	443
Names of the BRCONNECT Detail Logs	443
BRCONNECT Detail Log	444
BRCONNECT Summary Log	444
25.6 BRTOOLS	444
25.7 Initialization Profile init<DBSID>.sap	445
25.7.1 archive_copy_dir	445
25.7.2 archive_function	446
25.7.3 archive_stage_dir	447
25.7.4 backup_dev_type	447
25.7.5 backup_mode	449
25.7.6 backup_root_dir	450
25.7.7 backup_type	450
25.7.8 check_exclude	451
25.7.9 check_owner	452
25.7.10 cleanup_brarchive_log	452
25.7.11 cleanup_brbackup_log	452
25.7.12 cleanup_brconnect_log	452
25.7.13 cleanup_brrestore_log	452
25.7.14 cleanup_check_msg	452
25.7.15 cleanup_db_log	453
25.7.16 cleanup_disk_archive	453
25.7.17 cleanup_disk_backup	453
25.7.18 cleanup_exp_dump	453
25.7.19 cleanup_ora_trace	453
25.7.20 cleanup_owner	453
25.7.21 cleanup_sapdba_log	454
25.7.22 cleanup_xdb_log	454
25.7.23 compress	454
25.7.24 compress_cmd	455
25.7.25 compress_dir	455
25.7.26 copy_in_cmd	456
25.7.27 copy_out_cmd	456
25.7.28 cpio_disk_flags	456
25.7.29 cpio_flags	457
25.7.30 cpio_in_flags	457
25.7.31 dd_flags	457
25.7.32 dd_in_flags	457
25.7.33 disk_copy_cmd	458
25.7.34 dismount_cmd	458
25.7.35 exec_parallel	459
25.7.36 expir_period	459

25.7.37 mount_cmd	459
25.7.38 mount_par_file	460
25.7.39 new_db_home	461
25.7.40 next_exclude	461
25.7.41 next_limit_count	461
25.7.42 next_max_size	461
25.7.43 next_owner	462
25.7.44 next_special	462
25.7.45 next_table	462
25.7.46 orig_db_home	463
25.7.47 parallel_instances	463
25.7.48 primary_db	463
25.7.49 remote_host	464
25.7.50 remote_user	464
25.7.51 restore_mode	464
25.7.52 resync_cmd	465
25.7.53 rewind	466
25.7.54 rewind_offline	466
25.7.55 rman_channels	467
25.7.56 rman_diskratio	467
25.7.57 rman_duplex	467
25.7.58 rman_filesperset	467
25.7.59 rman_kbytes	468
25.7.60 rman_maxopenfiles	468
25.7.61 rman_parms	468
25.7.62 rman_pool	468
25.7.63 rman_proxy	469
25.7.64 rman_readrate	469
25.7.65 rman_setsize	469
25.7.66 saveset_members	469
25.7.67 split_cmd	470
25.7.68 stage_copy_cmd	470
25.7.69 stage_db_home	471
25.7.70 stage_root_dir	471
25.7.71 stats_bucket_count	471
25.7.72 stats_change_threshold	471
25.7.73 stats_dbms_stats	472
25.7.74 stats_exclude	472
25.7.75 stats_info_cubes	473
25.7.76 stats_limit_time	473
25.7.77 stats_method	473
25.7.78 stats_owner	474
25.7.79 stats_parallel_degree	474
25.7.80 stats_sample_size	474
25.7.81 stats_table	475
25.7.82 tape_address	475
25.7.83 tape_address_arch	476
25.7.84 tape_address_ctl	476
25.7.85 tape_address_ctl_arch	476

25.7.86 tape_address_rew	477
25.7.87 tape_address_rew_arch	477
25.7.88 tape_copy_cmd	477
25.7.89 tape_pos_cmd	479
25.7.90 tape_size	479
25.7.91 tape_size_arch	479
25.7.92 tape_use_count	480
25.7.93 uncompress_cmd	480
25.7.94 util_par_file	480
25.7.95 volume_archive	480
25.7.96 volume_backup	481
25.8 Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT	482
25.8.1 Log Types	482
25.8.2 File System Logs	483
25.8.3 Database Logs for BRBACKUP, BRARCHIVE, and BRCONNECT	483
25.8.4 Log Supplements	484
25.9 Messages and Return Codes for the BR Tools	485
26 Database Recovery with SQLPLUS	487
26.1 Types of Database Errors	487
26.1.1 Error Analysis	488
26.2 Recovery after User Errors	489
26.3 Recovery after Statement Errors	489
26.4 Recovery after Process Errors	489
26.5 Recovery after an Instance Error	490
26.6 Recovery after Media Errors	490
26.7 Recovery: One Control File Missing	491
26.8 Recovery: All Control Files Missing	492
26.9 Recovery: Current Online Redo Log Missing	495
26.10 Recovery: One Inactive Online Redo Log Missing	496
26.11 Recovery: User Tablespace Missing	497
26.12 Recovery: SYSTEM Tablespace Missing	499
26.13 Recovery: Index Tablespace Missing	500
26.14 Recovery: Errors During the Archiving of Online Redo Logs	501
26.15 Partial Recovery	502
26.15.1 Actions after a Partial Recovery	504
26.16 Automating the Recovery	506
26.17 Updating the Control File	506
27 The SAP Utilities with Windows NT	508
27.1 SAP Conventions (Windows NT)	508
27.1.1 Naming Conventions for Files (Windows NT)	508
27.1.2 Executables	509
27.1.3 Starting the SAP Utility Programs	509
27.1.4 Reorganization (Windows NT)	510
27.1.5 Database Analysis	510
27.1.6 Restrictions on SAPDBA (Windows NT)	511
27.2 Backup Strategy (Windows NT)	511
27.2.1 NTBackup	512
27.2.2 BRBACKUP/BRARCHIVE	512
27.2.3 Other Backup Programs	513
27.2.4 Structure-Retaining Database Copy or Restore on Windows NT	513
27.2.5 Offline Backup with Oracle Fail Safe for Cluster Systems	513



SAP Database Guide: Oracle (BC-DB-ORA-DBA)

Purpose

This component lets you administer your Oracle database with the SAP System. Read this documentation to make sure that you administer your database as efficiently as possible, which helps your company get the most from its SAP System.

SAP provides the following database administration (DBA) tools for this purpose:

- [BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT \[Page 356\]](#)
- [SAPDBA \[Page 184\]](#)



BRCONNECT was originally only a utility tool. Now you can also start it from the command line and it has taken over some of the functionality of SAPDBA. Therefore, certain SAPDBA functions are no longer being developed. We **strongly recommend** you to use BRCONNECT rather than SAPDBA where the two tools perform the same tasks.

Implementation Considerations

For more information if you are new to Oracle database administration with the SAP System, see [Getting Started with Oracle and the SAP System \[Page 23\]](#).

For more information about installing the Oracle database with an SAP System, see the documentation at <http://service.sap.com/instguides>.

Features

SAP DBA tools can perform the following important functions:

- [Backup, restore, and recovery \[Page 73\]](#)
- [Database system check with BRCONNECT \[Page 409\]](#)
- [Reorganization with SAPDBA \[Page 242\]](#)
- [Update statistics for the cost-based optimizer \[Page 420\]](#)

For more information about new features in this release, see [New Features in SAP Release 6.20 \[Page 16\]](#).

Constraints



Be sure to use the SAP tools SAPDBA, BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT in accordance with your conditions of use of the SAP System and other SAP products.



New Features

This section describes new features for Oracle database administration with SAPDBA.

1 New Features in SAP Release 6.20

There are the following new features in SAP Web Application Server (SAP Web AS) Release 6.20:

- Support for Oracle 9i:
 - [SAPDBA \[Page 184\]](#) and the [BR tools \[Page 356\]](#) now support Oracle 9i.
- RMAN backup without BACKINT:
 - [RMAN Backup with an External Backup Library \[Page 144\]](#)
 - SAP Note **420698**
- Update statistics for partitioned tables with BRCONNECT:
 - [Update Statistics with BRCONNECT \[Page 420\]](#)
 - SAP Note **424243**
- Update statistics for InfoCube tables with BRCONNECT:
 - [Update Statistics for InfoCube Tables \[Page 425\]](#)
 - SAP Note **428212**
- BRCONNECT support for Oracle monitoring in SAP Transaction RZ20:
 - [Monitoring the Oracle Database \[Ext.\]](#), especially [Database Health Alerts \[Ext.\]](#)
 - SAP Note **483659**
- New command options and initialization profile parameters
 - [Command Options for BRCONNECT \[Page 429\]](#):
 - [-o|output \[Page 441\]](#): new option process
 - [-f stats \[Page 436\]](#): new options `-f nocasc` and `-v index_store|cascade_store`
 - [-f next \[Page 435\]](#): new option `-f nocasc`
 - Initialization profile parameters:
 - [check_exclude \[Page 451\]](#): new values `non_sap` and `all_part`
 - [cleanup_check_msg \[Page 452\]](#)
 - [next_exclude \[Page 461\]](#): new value `all_part`
 - [next_special \[Page 462\]](#): new value `all_sel`
 - [stats_dbms_stats \[Page 472\]](#)
 - [stats_table \[Page 475\]](#): new value `all_ind`
 - SAP Notes **419679**, **424239**, **445884**, and **483639**.
- SAPDBA support for online reorganization of single tables

SAPDBA now supports online reorganization based on the Oracle internal PL/SQL functions. For more information, see [Reorganizing a Single Table or Index \[Page 260\]](#).
- SAPDBA support for LOB columns

SAPDBA now supports the reorganization of tables with all types of large object (LOB) columns. Large objects are recreated with the same physical characteristics as before the reorganization.

2 New Features in SAP Release 6.10

There are the following new features in SAP Release 6.10:

- [BRARCHIVE \[Page 383\]](#) support for [backup of offline redo log files \[Page 149\]](#) with Oracle Recovery Manager (RMAN)
 This enables you to develop a comprehensive strategy for database files and offline redo log files. You can also take advantage of internal block consistency checking by RMAN for offline redo log files.
- [Software compression \[Page 102\]](#) for backups on remote disk
 This helps to reduce the network traffic load required for backups. For more information, see [backup_dev_type \[Page 447\]](#).
- Support of `util_file_online` logic for [offline backups \[Page 81\]](#)
 This enables you to fully implement [split-mirror \[Page 170\]](#) and snapshot scenarios in BACKINT.
- New user interface for [BRCONNECT \[Page 407\]](#)
 The main new functions are:
 - [Database system check \[Page 409\]](#)
 - [Adapt next extents \[Page 419\]](#)
 - [Update statistics \[Page 420\]](#)
 - [Clean up old logs and traces \[Page 427\]](#)
 For more information about the new commands, see [Command Options for BRCONNECT \[Page 429\]](#).
- Support for the following new features in SAPDBA:
 - Partitioned objects – for more information, see:
 - [Reorganization with SAPDBA \[Page 242\]](#)
 - [Export and Import of Database Objects with SAPDBA \[Page 285\]](#)
 - [Reorganization \[Page 242\]](#) of index-organized tables
 - Creation and extension of locally managed (that is, "bitmap") tablespaces and reorganization of locally managed tablespaces.
 - New "rebuild online" option for index reorganization
 - Reorganization of new index types, bitmap and reverse – for more information, see [Reorganizing a Single Table or Index \[Page 260\]](#)
 - [Veritas Quick I/O \[Page 179\]](#) to administer files from file systems as if they were raw devices

3 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 \[Ext.\]](#).

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 \[Page 222\]](#).

New Support Package Concept

Read the Notes:

- 126769 for corrections in SAPDBA and the BR programs (for all releases)
- 141999 for new SAPDBA features (for all releases)

4 New Features in SAP R/3 Release 4.5

[SAPDBA \[Page 20\]](#)

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

4.1 Release 4.5: SAPDBA

- [Reorganization \[Page 259\]](#)
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.
- [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#)
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 \[Page 192\]](#).
- New Command Option
Use the command option `-v` | `-VERSION` to obtain patch information.

4.2 Release 4.5: BRBACKUP, BRARCHIVE, and BRRESTORE

- [Integration of the Recovery Managers into BRBACKUP/ BRRESTORE \[Page 137\] \[Page 83\]](#)
You can use the Oracle Recovery Manager (RMAN) to make an [incremental backup \[Page 83\]](#) 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*/ make this secure; you do not need to verify with `-verify`.

4.2 Release 4.5: BRBACKUP, BRARCHIVE, and BRRESTORE

- **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 \[Page 109\]](#) 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 \[Page 365\]](#)
For BRARCHIVE see [-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cds \[Page 390\]](#)

5 New Features in SAP R/3 Release 4.0

- [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\)](#)
[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 \[Page 344\]](#), [Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#).

- [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 \[Page 82\]](#)
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 \[Page 109\]](#) 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 \[Page 362\]](#) and [Completion of BRRESTORE Runs \[Page 396\]](#)
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 Verify \[Page 118\]](#)
You can check the Oracle block structure of restored backups or data files (online) with ORACLE DBVERIFY (*DB Check/ Verification – DB verification using DBVERIFY*). You can also verify backup tapes at a later point in time (*Backup database - backup function - Verify BRBACKUP tape*).
- [Reorganization Methods \[Page 284\]](#)
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 [Reorganization \[Page 259\]](#)
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.
- *Object handling - HideTab* in [Reorganization \[Page 259\]](#)
You can temporarily rename the tables you are reorganizing. This stops these tables being accessed during the reorganization process.



Getting Started with Oracle and the SAP System

Purpose

This section gives you an overview of database administration for Oracle databases running with the SAP System. The aim is to help you get started as quickly as possible by giving you concise information and pointers to further details.

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

Process Flow



You do not have to follow the sequence below rigidly. It is only a suggestion. However, be sure to consider **all** the items listed.

1. You read the Oracle documentation thoroughly.
2. You read the Release Notes that appear with each new SAP Release. To do this, choose *Help* → *Release Notes* in the SAP System.
See also [New Features \[Page 15\]](#).
3. You read the sections of the SAP Library on the [Computing Center Management System \[Page 24\]](#) to learn about how you can monitor a database and schedule and make backups.
4. You read the sections of the SAP Library on the ABAP Dictionary to learn about the conditions for creating [tables in the ABAP Dictionary \[Page 25\]](#) and on the database.
5. You read about the [Oracle Database Storage Parameters in the ABAP Dictionary \[Page 26\]](#).
6. If you have a problem, you use [SAP Service Marketplace \[Page 28\]](#) for support.
7. You consider using the [GoingLive and EarlyWatch \[Page 29\]](#) services.
8. If you intend to use raw devices, see [Oracle Databases on Raw Devices \[Page 33\]](#).
9. You take note of the [Limitations of the Database System \[Page 36\]](#).

Result

Now you are ready to [configure the database system \[Page 38\]](#).

6 Computing Center Management System

Use

You can use the Computing Center Management System (CCMS) to administer your Oracle database. For more information, see [SAP/Oracle DBA in CCMS \[Ext.\]](#).

Features

You can use CCMS to:

- Schedule a range of database administration activities – such as database backup, backup of the offline redo log files – using the [Database Planning Calendar. \[Ext.\]](#) You can choose from a range of action patterns that include the most commonly needed activities.
- [Display backup logs and status \[Ext.\]](#)
- [Update statistics for the cost-based optimizer \[Ext.\]](#)
- [Check the database system \[Ext.\]](#)
- [Monitor database operations \[Ext.\]](#)
- [Monitor database alerts \[Ext.\]](#)

7 Specification of SAP Tables in the ABAP Dictionary

Purpose

As soon as you create a database table for the SAP System, you can influence its storage parameters by maintaining its technical configuration in the [ABAP dictionary \[Ext.\]](#). This technical configuration is used to optimize the space requirements and access response of individual tables.

For more information, see [Creating Tables \[Ext.\]](#).

Process Flow

We recommend you to always maintain at least the parameters [Data Class \[Ext.\]](#) and [Size Category \[Ext.\]](#) 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 is automatically assigned to the correct database area when it is created. The F1 help for data class provides information on how to select the proper value for the table.

Data class is assigned to tablespace (for data) using the `TAORA` table. Data class is assigned to tablespace (for indexes) using the `IAORA` table.

- *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 in the database. If you need more space later, storage space is added corresponding to the selected category (*NEXT* extent). The F4 help on size category displays the number of data records that will fit in the assigned storage area of the database for each of the categories. A maximum of 300 extents ought to be enough for storing table contents, assuming a database block size of 8 KB.

Size category is assigned to extent size using an entry in the table `DD09L` and an analysis of the `TGORA` table for tables or `IGORA` for indexes. See also [Default Value for NEXT \[Page 257\]](#). 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 [sapdba -next <tablespace\(s\)> \[Page 351\]](#).

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

8 Oracle Database Storage Parameters in the ABAP Dictionary

Use

The [database utility \[Ext.\]](#) is the interface between the ABAP Dictionary and the relational database underlying the SAP System. You can use it to create, convert, and delete ABAP Dictionary objects, as well as database table, database views, and other SAP 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, such as `INITIAL EXTENT`, `NEXT EXTENT`, `MINIMUM EXTENTS`, `MAXIMUM EXTENTS`, `TABLESPACE`, `FREELIST GROUPS`, `FREELISTS`, `PCT FREE`, `PCT USED`, `INDEX ORGANIZED`, `PARTITION`, `COLUMN LIST`. 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. However, you cannot change all the values immediately. You can only change `MAXEXTENTS`, `NEXT`, `PCTFREE`, and `PCTUSED` immediately (these values are valid if a new storage area is requested for the object).

Prerequisites

To use the database utility, you need an authorization for authorization object `S_DDIC_OBJ`, such as `S_DDIC_ALL`. Make sure that the authorizations in your system are set so that only the database administrator is authorized to configure the database parameters. Check the settings and change them if necessary.

Features

Note the following parameters:

- `INDEX ORGANIZED`

This parameter lets you build a table in the same way as an index (that is, using a b*tree), so saving space.

The **advantages** of index-organized tables are:

- Less storage, since data only stored in b*tree instead of in table and index
- Rowid not stored in index entry of b*tree
- Faster key-based access to table data

The **disadvantages** of index-organized tables are:

- `UNIQUE` constraints not allowed
- Cannot be stored in a cluster
- Cannot contain `LONG` columns (but `LOB` columns are possible)

- `PARTITION BY`

This allows you to partition tables by using a range expression.

The **advantages** of partitioned tables are:

- Logical attributes (such as table or indexed columns, constraints) are same in all partitions
- Physical attributes (tablespace, storage parameters) might differ in the partitions
- Data of partitioned objects can be handled in the same way as unpartitioned tables
- Partitions of a table can be separately exported, imported, dropped, set offline, backed up, and so on
- Queries can be performed only on specific partitions

The **disadvantages** of partitioned tables are:

- Bitmap indexes on partitioned tables can only be local
- Rule-based optimizer is not available for partitions

For more information on the other parameters, see the Oracle documentation.

Activities

For more information on how to call the database utility and set storage parameters, see:

- [Database Utility \[Ext.\]](#)
- [Storage Parameters \[Ext.\]](#)

9 Support with SAP Service Marketplace

Purpose

If problem situations occur in the system, SAP Service Marketplace provides fast, effective help. Here you can address questions to SAP directly and immediately receive an initial response. You receive the same information that SAP itself uses for support work.

SAP support staff use incoming customer messages to write SAP Notes. SAP developers also create Notes to help you fix potential problems or to offer missing information.

Process Flow

1. You sign on to SAP Service Marketplace using `http://service.sap.com`.
2. You use the alias `message` to enter your problem message
The system searches automatically for SAP Notes matching the words in your problem message.
3. You can also use the alias `notes` to reach the SAP Notes area, where you can search manually for SAP Notes to help fix your problem. There are different ways to search for helpful notes, such as:
 - Entering the application area (for example, BC-DB-ORA)
 - Entering the SAP Release
 - Specifying the number of the note, if known
 - Searching with free text, by entering a meaningful keyword, as in the following examples.



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

10 GoingLive and EarlyWatch

Purpose

This section describes the GoingLive and EarlyWatch services, which together provide a proactive service for your SAP System. The benefits are:

- Prevention of bottlenecks
- Early detection of potential problems
- Optimized performance and system availability

These benefits are achieved by:

- Certified and trained SAP experts with in-depth knowledge of the R/3 System
- Best practices database with up-to-date information
- Service procedures used to maintain top service performance
- Analysis tools used to expose system weak points
- Empowering workshops to deliver expertise to your experts

Whereas GoingLive supports you during the implementation phase, EarlyWatch supports you doing ongoing live production. By using these services, you can make sure that your system is optimally tuned and less likely to fail. The result is improved system availability.

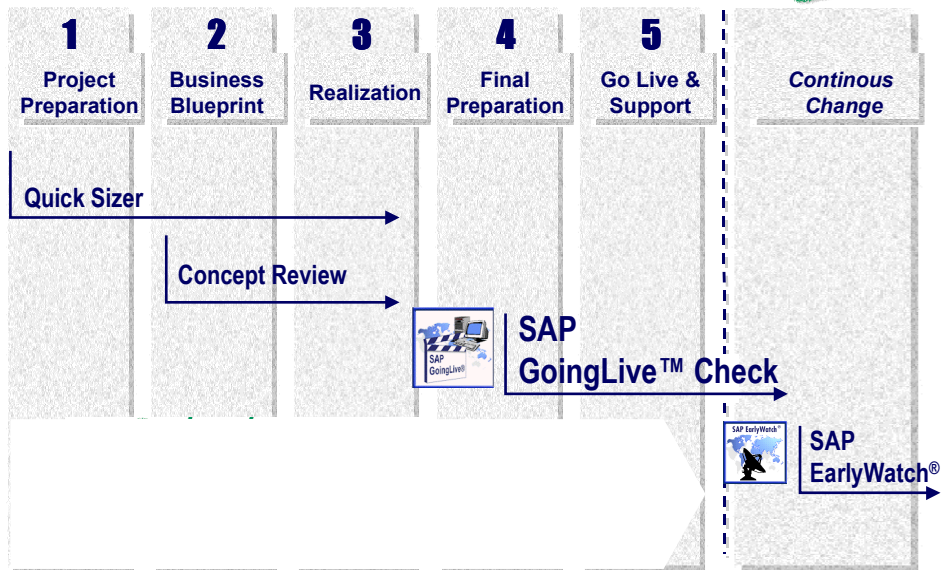
For more information, enter the following in your browser:

<http://service.sap.com/earlywatch>

Process Flow

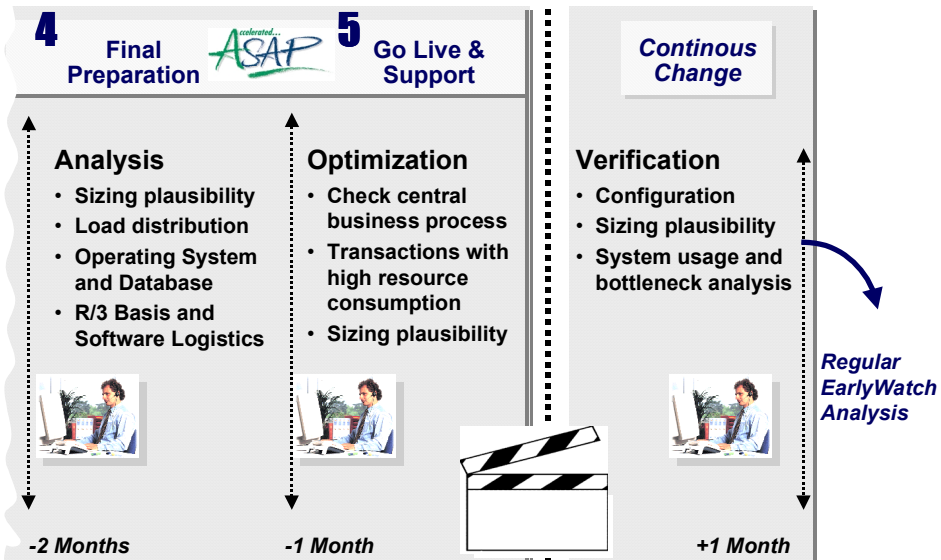
Both services contribute to improving the availability of your system because they stress prevention of problems before they become serious enough to cause downtime. As the following diagram shows, these services are complementary, with EarlyWatch taking over once your system has gone live:

Services and Tools provided by SAP *ASAP*



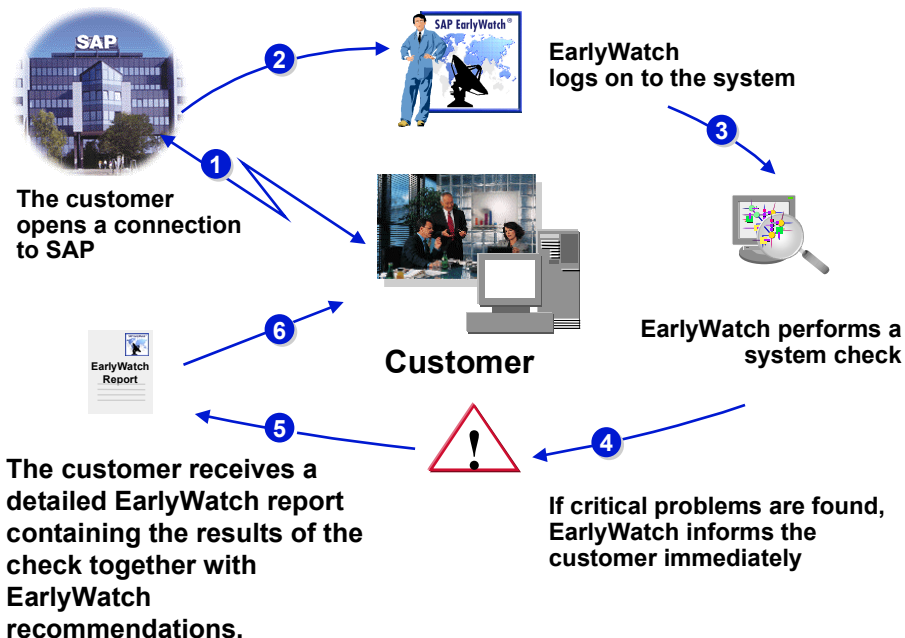
The most critical phase in achieving high availability for your R/3 System is when you are going live, because it is more difficult to change a system once it is in production. The following diagram shows the different ways that GoingLive can help you during this critical phase and how EarlyWatch can help afterwards:

GoingLive™ Check Service Session



The EarlyWatch service works as follows:

How Does EarlyWatch Work?



EarlyWatch focuses on the following aspects:

- Server analysis

- Database analysis
- Configuration analysis
- Application analysis
- Workload analysis

EarlyWatch Alert – a free part of your standard maintenance contract with SAP – is a preventive service designed to help you take rapid action before potential problems can lead to actual downtime. In addition to EarlyWatch Alert, you can also decide to have an EarlyWatch session for a more detailed analysis of your system.

Result

You receive a service report after an EarlyWatch session, broken down into the following sections:

The EarlyWatch Service Report

Chapter 1 Global EW Summary (red, yellow, green)



Chapter 2 Global Analysis

Chapter 3 Client/Server Analysis

Chapter 4 Application Server Analysis (Server 1 - N)



Chapter 5 Database Server Analysis

Chapter 6 Summary and Recommendations

CPU	
Memory	
Swap Space	
Log Statistics	
Parameter Changes	
Buffer Statistics	

Chapter 7 Appendix A: Checklists

Chapter 8 Appendix B: Glossary

11 Oracle Databases on Raw Devices

Use

You can operate an Oracle database on raw devices. This means that the Oracle database management system avoids the file management of the UNIX system and instead writes data directly to a disk partition. Therefore, the partition is used as a raw device and does not contain a file system. This type of storage improves the speed of data access, but requires its own file management. For more information, see [Raw Devices and SAPDBA \[Page 33\]](#).

The SAP utilities for database backup support raw devices. For more information, see [Raw Devices with BRBACKUP/ BRRESTORE \[Page 34\]](#).



Raw devices are only used with the UNIX operating system, **not** with Windows NT.

Features

There are the following advantages and disadvantages when using raw devices.

Advantages

Working with raw devices improves performance for the following reasons:

- Since the usual buffer cache for a file system is not needed, faster data access is possible and 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.

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 be aware of the disadvantages (for example, no end-of-media handling).



SAPDBA can make only restricted checks on raw devices, because it can check only those devices that belong to the database user `ora<sapsid>`. Raw devices are recognized only if they are allocated to the database `DBSID`. SAPDBA does not check whether there are also objects from other applications on these raw devices.

11.1 Raw Devices and SAPDBA

11.2 Raw Devices with BRBACKUP and BRRESTORE

The following information is important if you intend to use SAPDBA for Oracle with raw devices:

- Be sure to observe the SAP naming convention for tablespaces on raw devices:

`<SAPDATA_HOME>/sapraw/<TSP>_<Number> → <Raw-Device-Dir>/<Device>`

This is composed as follows:

- `<Raw-Device-Dir>` specifies the central directory containing the devices. This directory can also be specified in the [rawdev_dir \[Page 202\]](#) parameter.
- `<Device>` consists of `<DBSID><TSP>_<Number>`.
- `<Number>` is the sequence number assigned to the raw devices (or files) belonging to the tablespace.

See also [Environment Variables \(UNIX\) \[Page 51\]](#).

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
```

Compare this with the [SAP naming conventions for tablespaces \[Page 49\]](#) in the file system.

- The database link structure is 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 \[Page 206\]](#).
- Each raw device can contain only one tablespace file (this is an Oracle requirement). 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 \[Page 233\]](#).
- It might be sensible to specify the [rawdev_dir \[Page 202\]](#) and [blkdev_dir \[Page 199\]](#) parameters in the `init<DBSID>.dba` profile.

11.2 Raw Devices with BRBACKUP and 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 functional scope of the programs. The known functionality of BRBACKUP and BRRESTORE for backing up and restoring file systems remains unchanged.

BRBACKUP and BRRESTORE use the `dd` command to access raw devices:

- With `dd`, you can write directly from the raw device to tape or to the raw device from tape (that is, backup to tape or restore from tape) and you can do the same to or from disk.
- For a backup with software compression, `dd` output is sent directly to the `compress`. For a restore with decompression, output of `uncompress` is sent to `dd`.

11.2 Raw Devices with BRBACKUP and BRRESTORE

- For a backup on a remote computer, `dd` output is sent, for example, directly to `rsh` (`/remsh` and so on). For a restore from a remote computer, it is sent from `rsh` (`/remsh` and so on) to the `dd` command.
- You can define options for `dd` command using the `init<DBSID>.sap` parameters [dd flags \[Page 457\]](#) and [dd in flags \[Page 457\]](#).



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 are necessary for backup of the next database files.

12 Limitations of the Oracle Database System

Definition

The Oracle database has the limitations described in this section. The parameters `MAXEXTENTS`, `DBFILES`, and `MAXDATAFILES` are discussed.

Use

You need to be aware of the limitations described below when you are:

- [Reorganizing the database \[Page 242\]](#)
- [Checking a tablespace \[Page 230\]](#) before possibly extending it

Structure

- Maximum number of extents per table or index – `MAXEXTENTS`

- Soft limit

The SAP software value for `MAXEXTENTS` is 100.

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, we recommend you to [change the extent parameters \[Page 254\]](#) as follows so that the system can continue running:

- i. Increase `MAXEXTENTS`.
- ii. Increase `NEXT`, so that the system does not reach the increased maximum number of extents in the near future. For more information on how to do this from the command line, see [sapdba -next <tablespace\(s\)> \[Page 351\]](#).

Only reorganize the object in an emergency.

- Hard limit

The usual hardware value for `MAXEXTENTS` is 505.

This limit only exists for Oracle releases **before 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 (see the table below). If you reach the hardware limit you **must** reorganize the affected table or index.

The following table shows the hard and soft limits for `MAXEXTENTS` by block size:

Block Size and `MAXEXTENTS`

Block size in KB	Hard limit for <code>MAXEXTENTS</code>	Soft limit for <code>MAXEXTENT S</code>
2	121	100
4	249	100
8	505	100



The software value of `MAXEXTENTS = 100` enables you to anticipate most cases where the number of extents is too high. This prevents you from reaching the hard limit and avoids reorganization.

- Maximum number of files per database – `DB_FILES`

- Soft limit

The SAP software value for `DB_FILES` is 254.

The database system only supports a specific number of data files in the database, depending on the host system, and this is specified by the `DB_FILES` parameter in the `init<DBSID>.ora` profile. If your database approaches this limit, you can reduce the number of data files by reorganizing tablespaces that have more than one file. However, this is **not** likely to occur with a limit of 254 files.

- Hard limit

The hard limit for `DB_FILES` depends on the operating system but is at least 1022.

`DB_FILES` can be increased to the value of `MAXDATAFILES`, the value of which was specified when the database was created. `MAXDATAFILES` itself must be less than the permissible maximum number of open files supported by the operating system. The default value for `MAXDATAFILES` is also 254.



Do **not** regularly reorganize the database to reduce the number of data files. This is because the possible number of data files is large, so is not normally reached.

13 Database System Configuration

Purpose

This section tells you how to configure your Oracle database system.

Prerequisites

During database installation, which is not covered in this documentation, you need to [change the initial database user passwords \[Page 39\]](#) and [initialize SAP connect \[Page 40\]](#).

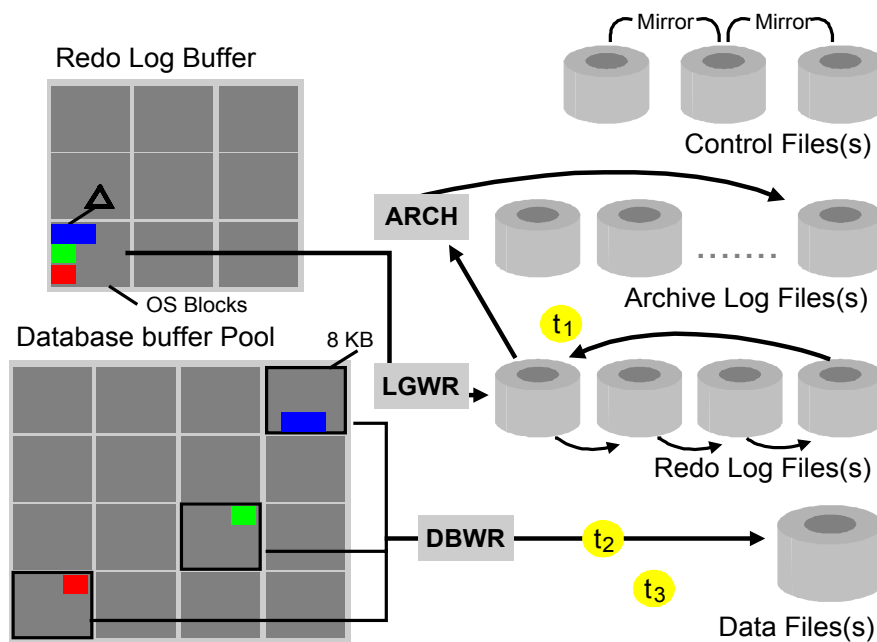
Process Flow

1. You set up archiving, making sure that the database runs in ARCHIVELOG mode and with automatic archival enabled.
[See Setting Up Archiving \[Page 43\]](#) and [Setting Archive and Database Mode with SAPDBA \[Page 41\]](#).
2. You mirror the control files on separate disks.
[See Mirroring the Control File \[Page 45\]](#).
3. You mirror the online redo log files on separate disks (at operating system level and/or using Oracle resources). With Oracle support, you can set this up when you install the database system.
[See Mirroring Online Redo Log Files \[Page 45\]](#).
4. You install online redo log files, offline redo logs files (that is, online redo log files saved to the archive directory), and data files to separate disks.
[See Organizing Disk Storage \[Page 46\]](#).
5. You make sure that the block sizes of the database system and the operating system are the same for security and performance reasons. The block size of the operating system is reformatted to 8K during the installation of the SAP System.

The AIX operating system is an exception, because the block size of the database system must be 8 KB and the block size of the operating system must be 4 KB.
6. You familiarize yourself with the tablespace structure of the SAP System, being sure to monitor the critical tablespaces.
[See SAP Naming Conventions for Tablespaces and Data Files \[Page 49\]](#) and the graphic below.

Structure of Oracle database

13.1 Changing Database User Passwords



7. You familiarize yourself with environment variables and set them as required. Refer to:
 - [Environment Variables \(UNIX\) \[Page 51\]](#)
 - [Environment Variables \(Windows NT\) \[Page 53\]](#)
8. You familiarize yourself with the directory structure. Refer to:
 - [Directory Structure \(UNIX\) \[Page 55\]](#)
 - [Directory Structure \(Windows NT\) \[Page 60\]](#)

13.1 Changing Database User Passwords

Use

You can use SAPDBA for Oracle to display and change the password of database users, such as `SAPR3` or `SAP<SAPSID>`. You can do this with the SAPDBA menus, as described here, or with the [SAPDBA command `alter user` \[Page 345\]](#).

You can also display information about the current operating system and database users with the SAPDBA option *User and Security*. SAPDBA tells you if the database user passwords are still set to the default values.



Do not leave the passwords for the database user set to the default values.

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

Prerequisites

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

13.2 Initializing SAP Connect

1. Choose *User and security* → *Change password*.
2. Select *Password entering mode*, either *invisible* or *visible*.
SAP displays the default user, *SAPR3*.
3. Enter the *User* for which you want to change the password.
4. Choose *Change password*.
5. Enter the new password as requested by SAPDBA
The new password must be no longer than eight characters.

Result

SAPDBA stores the new password as follows:

User	Table	Encryption (from Release 4.5B)
SAPR3	"OPS\$<ORACLE_SID>ADM".SAPUSER	Stored as SAPR3-CRYPT
SAP<SAPSID>	"OPS\$<SAPSID>ADM".SAPUSER	Stored as SAP<SAPSID>-CRYPT



In the case of the SAP<SAPSID> user, <SAPSID> refers to the SAP System ID. For example, if the SAP System ID is C11, the SAP<SAPSID> user is SAPC11, stored in the table OPS\$C11ADM.SAPUSER. The first character of <SAPSID> must be a letter and there is no distinction between uppercase and lowercase.
Database users other than SAPR3 and SAP<SAPSID> are **not** stored in SAPUSER database tables.

13.2 Initializing SAP Connect

Use

You can use SAPDBA for Oracle to initialize SAP connect, which you normally do during database installation. You can do this with the SAPDBA menus, as described here, or with the [SAPDBA command init_sap_connect \[Page 351\]](#).

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- If you use the SAPDBA menus, you can set the `sapsid` switch (see below) when you start SAPDBA, as follows:

```
sapdba -sapsid <SID1>,<SID2>,...
```

Procedure

6. Choose *User and security* → *Initialize SAP Connect*.
SAPDBA displays what it does if you continue.
7. Choose whether you want to continue and initialize SAP connect.

Result

The result described below refers to both the command line and the menu execution of this function.

13.3 Setting Archive and Database Mode with SAPDBA

SAPDBA does the following when you initialize SAP connect:

- If `remote_os_authent` in the file `init<ORACLE_SID>.ora` is **not** set, SAPDBA sets it to `remote_os_authent = true`.
- SAPDBA performs a series of actions for the following database users, depending on whether the `sapsid` switch is set or not:
 - If `sapsid` is **not** set, for the `SAPR3` user and system ID `ORACLE_SID`
 - If `sapsid` **is** set, for the users `SAP<SID1>`, `SAP<SID2>` . . . and system IDs `<SID1>`, `<SID2>`

The system ID is represented by the variable `<SID>` below. Remember that, if the `sapsid` switch is set, there can be multiple users, schemas, and tables.

SAPDBA performs the following actions:

- If not already present, SAPDBA creates the users specified above, that is, either `SAPR3` or `SAP<SID1>`, `SAP<SID2>` . . . The initial passwords for the new users are set to `SAP`.
- SAPDBA deletes the schemas `OPS$<SID>ADM` and all associated tables (this should only affect the `SAPUSER` tables).
- SAPDBA recreates the schemas `OPS$<SID>ADM` without password, that is, they are identified at operating system level using the operating system user `<SID>ADM`.
- SAPDBA recreates the tables `OPS$<SID>ADM.SAPUSER` and enters the `SAPR3` or the `SAP<SID>` users in encrypted form (starting with Release 4.5B) under `SAPR3 - CRYPT` or `SAP<SID> - CRYPT` in the tables.

13.3 Setting Archive and Database Mode with SAPDBA

Use

You can use SAPDBA for Oracle to:

- Set the database mode and the archiving mode
- Show archive information and history
- Select threads and instances for the above operations

Prerequisites

- SAPDBA is installed correctly. Refer to [SAPDBA Configuration \[Page 185\]](#).
- You know how to use SAPDBA and have logged on. Refer to [Starting SAPDBA \[Page 193\]](#).

Procedure

1. Choose *Archive Mode* in SAPDBA.

The system displays the following:

```
LOCAL INSTANCE: C11
DATABASE LOG MODE: archivelog
```

13.3 Setting Archive and Database Mode with SAPDBA

SELECTED INSTANCE: C11
 INSTANCE STATE: opened
 AUTOMATIC ARCHIVAL: enabled

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



The database must normally run with DATABASE LOG MODE set to `archivelog` and AUTOMATIC ARCHIVAL set to enabled. For more information, see [Setting Up Archiving \[Page 43\]](#).

2. To change the database log mode, choose *Toggle database log mode*.
3. To enable or disable automatic archival, choose *Toggle automatic archival*.
4. Choose *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 COMPLETE 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)

If an error situation occurs in the database, making a recovery necessary, be sure to view this information before starting the recovery.

5. 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, the following information appears:
 - LOCAL INSTANCE: Name of the local instance (for example, 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

13.4 Setting Up Archiving

Use

This procedure tells you how to check the archiving parameters and ARCHIVELOG mode for your Oracle database and, if necessary, how to reset these. It is very important that:

- The database runs in ARCHIVELOG mode
- Automatic archival is enabled

After correct installation for an SAP System, the Oracle database meets both these criteria. In this case, the online redo log files are automatically archived when full (that is, following a redo log switch). This is important because it allows the online redo log files to be reused for fresh archive data, so that archiving of the log files can continue at all times.

Prerequisites

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

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

`log_archive_start = true` causes the background archive process ARCH to be started automatically when the database is started. This means that automatic archival mode is enabled.

`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 (this example is for a single instance installation on UNIX):

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

See SAP Note 316642 for information on how to avoid an "archiver stuck" error by changing the archive directory.

13.4 Setting Up Archiving

For more information, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).



The Oracle database names the offline redo log files using the string `<ORACLE_SID>arch` followed by the log sequence number. You can [back up the offline redo log files with SAPDBA \[Page 313\]](#) with `BRARCHIVE`. The offline redo log files are the copies of the online redo log files saved in the archive directory.

Procedure

1. Make sure that:

- The archive directory (under UNIX: `saparch`) exists.
- The directory is not write-protected.
- The directory has enough free space. Otherwise, the archiving process cannot archive any log files and no further actions are possible on the database (this is known as “Archiver Stuck”).



For more information on the SAPDBA commands mentioned below, see [Setting Archive and Database Mode with SAPDBA \[Page 41\]](#).

2. Check ARCHIVELOG mode by choosing *Archive Mode* in SAPDBA.

The system displays archiving details. Database log mode should be set to `archivelog`.

3. If you need to reset ARCHIVELOG mode, do this in either of the following ways:

- Choose *Archive mode* → *Toggle database log mode* in SAPDBA.
SAPDBA shuts down the database, configures the database mode, and restarts the database.
- Enter the following commands in the Oracle tool SQLPLUS:

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

SQLPLUS starts the database in ARCHIVELOG mode.

Repeat the previous step to check that ARCHIVELOG has indeed been set correctly.

4. Check whether AUTOMATIC ARCHIVAL is enabled by choosing *Archive Mode* in SAPDBA.

The system displays archiving details. Automatic Archival should be set to `enabled`.

5. If you need to enable AUTOMATIC ARCHIVAL (that is, to start the ARCH process), do this in either of the following ways:

- Choose *Archive mode* → *Toggle automatic archival* in SAPDBA.
SAPDBA enables automatic archival by starting the ARCH process.
- Enter the following commands in the Oracle tool SQLPLUS:

```
SQLPLUS> connect internal
SQLPLUS> alter system archive log start;
SQLPLUS> archive log list
```

Repeat the previous step to check that AUTOMATIC ARCHIVAL has indeed been enabled.

13.5 Mirroring the Control File

Use

It is essential to mirror the control file in your Oracle database.

If you only have one copy of the control file and lose this copy due to a disk error or other problems, it is probably impossible to completely recover the database. The result is inevitable data loss.

To prevent this happening, create multiple copies of the control file. When the SAP 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

If necessary, change the standard mirroring of the control file by setting the `control_files` parameter.



Here is a sample entry in profile `init<DBSID>.ora`:

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

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

13.6 Mirroring Online Redo Log Files

Use

It is essential to mirror the online redo log files in your Oracle database.

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, we strongly recommend that you mirror the online redo log files. We recommend that you make at least one copy.

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

13.7 Organizing Disk Storage

a case, you must recover the original mirroring of the online redo log files as quickly as possible.

For more information see:

- [Database Recovery with SQLPLUS \[Page 487\]](#)
- [Recovery: One Inactive Redo Log Missing \[Page 496\]](#)

Prerequisites

When an SAP 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 (that is, the group has two members).

Procedure

This procedure describes how to check the mirroring with SAPDBA.

1. During SAPDBA startup, any missing online redo log files are displayed if the profile parameters `check_initial_settings` or `check_redolog_mirror` are configured properly.
See [check initial settings \[Page 199\]](#)
2. Choose *Archive mode* → *Show all archive information: ONLINE REDO LOGS* in SAPDBA. Refer to [Setting Archive and Database Mode with SAPDBA \[Page 41\]](#).
3. When the database is open, you can also check the mirroring with the Oracle program SQLPLUS:


```
SQLPLUS> connect internal
SQLPLUS> select * from V$logfile;
```
4. 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 information, see your operating system documentation.

For more information on the security of offline redo log files, see [Backup of the Offline Redo Log Files \[Page 313\]](#).

13.7 Organizing Disk Storage

Use

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



Disk number	Directory	Contents of directory
1	origlogA	Online redo log files from the first and third group (Set A)
2	origlogB	Online redo log files from the second and fourth group (Set B)
3	mirrorlogA	Mirrored online redo log files from the first and third group (Set A)
4	mirrorlogB	Mirrored online redo log files from the second and fourth group (Set B)

5	sapdata1	Database files, mirror of the control file
6	sapdata2	Database files, mirror of the control file
7	sapdata<n>	Other database files in sapdata3 up to sapdata<n>, each on separate disks when possible



A disk assignment similar to this is **essential for reliable database operation**. For performance reasons, we recommend you to distribute the online redo log groups to four disks, as shown above.

Procedure

1. 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 across any number of disks.

2. Make sure that the offline redo log files (that is, online redo log files saved to the archive directory) are not stored on the same disk as the online redo log files.

The offline redo log files are **not** shown in the above example.

13.8 Database Parameters

Definition

The SAP System for the Oracle database comes with a standard initialization profile for the database parameters:

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

For example:

```
/oracle/C11/dbs/initC11.ora
```

Use

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

Structure

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. Make 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 45] .
log_archive_dest	OS> <code>?/saparch/<DBSID>arch</code> , where <code>?/saparch</code> can be a soft link to a separate disk or a separate mount point. See Setting Up Archiving [Page 43] .

13.8 Database Parameters

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 for the system.	One system installed on one host: $\text{ram} = 1.0 * \text{RAM}$ Two systems installed on one host: $\text{ram} = 0.5 * \text{RAM}$
rating Proportion of RAM taken up by the database and the SAP System	Default: rating=0.25 (25% of RAM reserved for the database, 75% for the SAP System) Database-only server: rating=0.8 (80% of RAM is reserved for the database) If $\text{ram} < \text{RAM} * 0.8$: rating=1.0 (100% of RAM is reserved for the database)
db_block_size db_block_size (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
shared_pool_size db_block_size (unit: bytes) buffer size for parsed SQL statements in the Oracle System Global Area (SGA)	$\text{shared_pool_size} = \text{ram} * \text{rating} * 0.4 * 1024 * 1024$ The factor 0.4 means that shared_pool_size occupies 40% of the calculated SGA. $\text{shared_pool_size} = 20.971.520$ if the value calculated using the formula above is lower than 20971520 Bytes.
db_block_buffers db_block_size (unit: blocks) buffer size for data blocks and index blocks in the Oracle SGA	$\text{db_block_buffers} = \text{ram} * \text{rating} * 0.6 * 1024 * 1024 / \text{db_block_size}$ The factor 0.6 means that db_block_buffers occupies 60% of the calculated SGA.
log_buffer db_block_size (unit: bytes) buffer size for redo log entries in the Oracle SGA	Default: log_buffer=327.680 Part of the SGA, which is very small when compared with shared_pool_size and db_block_buffers.

13.9 SAP Naming Conventions for Tablespaces and Data Files

row_cache_cursors	row_cache_cursors=
Number of cursors used for selecting lines from the ABAP Dictionary.	300, if ram>=256
	200, if ram<256
	100, if ram<128

13.9 SAP Naming Conventions for Tablespaces and Data Files

Definition

In SAP Systems with the Oracle database, tablespaces and data files are named according to the conventions described in this section.

Use

- You might need to [create a new tablespace \[Page 235\]](#) with SAPDBA, for example, in the following situations:
 - During the repository switch in a SAP system upgrade
 - When moving a table to a separate tablespace (for example, for administrative reasons)
- For more information on extending a tablespace, see [Extending a Tablespace \[Page 233\]](#).



We strongly recommend you to create new tablespaces and add data files to extend existing ones in accordance with the SAP naming conventions (see "Structure" below).

Monitoring Tablespaces

Monitor the following tablespaces with special care:

- Certain 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 SAP System:
 - PSAPSTABD/I
 - PSAPCLUD/I
 - PSAPBTABD/I
 - PSAPPOOLD/I
 - PSAPPROTD/I
- Certain tablespaces contain a large number of tables, which means that the number of extents for each tablespace is very high. You might experience problems with Oracle when you [reorganize special tablespaces \[Page 270\]](#).
 - PSAPSTABD/I
 - PSAPBTABD/I
 - PSAPPOOLD/I

13.9 SAP Naming Conventions for Tablespaces and Data Files

- The SAP software compresses a large portion of its tables in order to save disk space. Therefore, when you export data, there is no point in compressing the tablespaces that contain the tables listed below. SAP's compressed tables are held primarily in the following tablespaces:

- PSAPCLUD/I
- PSAPDOCUD/I
- PSAPPOOLD/I
- PSAPSOURCED/I or PSAPES<release-number>D/I
- PSAPLOADD/I or PSAPEL<release-number>D/I

Structure

Tablespaces

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
Oracle tablespaces	These tablespaces are required for operation of the Oracle DBMS, and contain no SAP data.
SYSTEM	Oracle DDIC
PSAPROLL	Rollback segments
PSAPTEMP	Sort processes
Basis tablespaces	
PSAPEL300D/I	Development environment loads
PSAPES300D/I	Development environment sources
PSAPLOADD/I	Screen and report loads (ABAP)
PSAPSOURCED/I	Screen and report sources (ABAP)
PSAPDDICD/I	ABAP Dictionary
PSAPPROTD/I	Log-like tables (such as spool)
Application	
PSAPCLUD/I	Cluster tables
PSAPPOOLD/I	Pooled tables (such as ATAB)
PSAPSTABD/I	Master data, transparent tables
PSAPBTABD/I	Transaction data, transparent tables
PSAPDOCUD/I	Doc., Sapscrip, Sapfind
Customers	

PSAPUSER1D/I	Customer tables
--------------	-----------------

Data Files

When you add a new data file to extend a tablespace, 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.

The naming conventions are as follows:

- Directory: <ORACLE_HOME>/sapdata<n>/<tablespace name>_<file number>
- File name: <tablespace name>.data<file number>

For more information on <ORACLE_HOME>, see:

- [Environment Variables \(UNIX\) \[Page 51\]](#)
- [Environment Variables \(Windows NT\) \[Page 53\]](#)

<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 that is created for each new file in a tablespace.

13.10 Environment Variables (UNIX)

Definition

Environment variables define parameter values used by the Oracle database and SAPDBA. This section describes the variables used when the operating system is UNIX.

Use

The database uses the parameter values for many different purposes. SAPDBA also uses the parameter values.

Structure

The following variables are required:

- ORACLE_SID
System ID of the database instance
Example: C11
SAPSID or sapsid refers to the 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.

13.10 Environment Variables (UNIX)

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/-statistics` logs.
Default value: `$SAPDATA_HOME/sapcheck`
- `SAPREORG`
Directory for all other SAPDBA logs, as well as shell and SQL scripts. It is also the standard directory for export dump files, if the parameter `exireo_dumpdir` in the profile `init<DBSID>.dba` is not set.
Default value: `$SAPDATA_HOME/sapreorg`
- `SAPTRACE`
Directory for Oracle trace files and the alert file.
Default value: `$SAPDATA_HOME/saptrace`
- `SAPDATA1`
Directory of the database data files.
Default value: `$SAPDATA_HOME/sapdata1`
(The same for `SAPDATA<n>`, `n=1,...99`).
The environment variables `SAPDATA<n>` must only be defined if directories are used that differ from the default.
- `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_TRACE

Setting the trace function for error analysis.

See Note 29324 for setting options.

See also:

[Environment Variables \(Windows NT\) \[Page 53\]](#)

13.11 Environment Variables (Windows NT)

Definition

Environment variables define parameter values used by the Oracle database and SAPDBA. This section describes the variables used when the operating system is Windows NT.

Use

The database uses the parameter values for many different purposes. SAPDBA also uses the parameter values.

Structure

The following variables are required:

- ORACLE_SID

System ID of the database instance

Example: C11

SAPSID or sapsid refers to the 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. This variable no longer needs to be set. If not set, the SAP tools determine its value from the registry.

Standard: <drive>:\orant

Example: D:\orant

This variable is optional with Oracle version 8.1 and later.

- SAPDATA_HOME

Directory of the database files.

Standard: <drive>:\oracle\<DBSID>

Example: E:\oracle\C11



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

13.11 Environment Variables (Windows NT)

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 BRCONNECT logs.
Default value: %SAPDATA_HOME%\sapcheck
- **SAPREORG**
Directory for SAPDBA logs, as well as shell and SQL scripts. It is also the standard directory for export 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 database data files.
Default value: %SAPDATA_HOME%\sapdata1
(The same for `SAPDATA<n>`, `n=1, ... 99`).



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. For 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

See also:

[Environment Variables \(UNIX\) \[Page 51\]](#)

13.12 Directory Structure (UNIX)

Definition

This section describes the directory structure for the Oracle database with the UNIX operating system.

Use

The directories contain a range of files such as profiles, log files, scripts, executables, and so on.

The following conventions apply to this section:

Term	Meaning
<code>\$<name></code>	Environment variable
<code><DBSID></code>	<code>\$ORACLE_SID</code>
<code><TSP></code>	Tablespace short name For example, BTABD for the tablespace PSAPBTABD.

Structure

There are the following main structures:

- `$ORACLE_HOME` for Oracle-specific objects with the default directory `/oracle/<DBSID>`
- `$SAPDATA_HOME` for SAP-specific objects with the default directory `/oracle/<DBSID>`
- Executables with the directory `/usr/sap/<SAPSID>/SYS/exe/run`

Oracle Home Directory

This directory contains the following subdirectories:

- `db/`
 - `init<DBSID>.ora` Profile for Oracle
 - `init<DBSID>.dba` Profile for SAPDBA
 - `init<DBSID>.sap` Profile for BRBACKUP, BRARCHIVE, and BRRESTORE
- `bin/` contains binaries such as `oracle`, `svrmgrl`, `sqlplus`, `exp`, `imp`
 - `rwsr-xr-x ora<dbsid> dba oracle`
 - `rwxr-xr-x ora<dbsid> dba svrmgrl`

SAP Home Directory

For the files starting `log_`, `g` means group and `m` means member:

- `origlogA/`
 - `log_g11m1.dbf`
 - `log_g13m1.dbf`
- `origlogB/`
 - `log_g11m1.dbf`

13.12 Directory Structure (UNIX)

- log_g13m1.dbf

Mirrored redo logs are optional – although we strongly recommend using them – so the specification for `mirrlogA` and `mirrlogB` might vary:

- `mirrlogA/`
 - log_g11m2.dbf
 - log_g13m2.dbf
- `mirrlogB/`
 - log_g12m2.dbf
 - log_g14m2.dbf
- `sapdata1/`
 - `cntrl/cntrl<DBSID>.dbf` Database control file
 - `system_1/system.data1`
 - `btabd_1/btabd.data1` Example of a SAP data file for tablespace PSAPBTABD
 - ...
- `sapdata2/`
 - `cntrl/cntrl<DBSID>.dbf` Database control file
 - `system_1/system.data1`
 - `btabi_1/btabi.data1` Example of a SAP data file for tablespace PSAPBTABI
 - ...
- `sapdata<n>/`
- `sapreorg/`

Logs written by SAPDBA:

- `<timestamp>.rch` Log check option
- `<timestamp>.ral` Alter storage in reorganization menu
- `<timestamp>.rsi` Reorganize a single table or index
- `<timestamp>.rtc` Reorganize tablespace without data file
- `<timestamp>.rtd` Reorganize tablespace with data file
- `<timestamp>.ext` Tablespace extension
- `<timestamp>.exs` Export table and index structures
- `<timestamp>.exd` Export table with data
- `<timestamp>.exg` Export constraints, synonyms, sequences, views, grants
- `<timestamp>.ims` Import structure
- `<timestamp>.imd` Import table with data
- `<timestamp>.rcv` Recovery
- `<timestamp>.brb` Backup start with BRBACKUP

- <timestamp>.bra Archive start with BRARCHIVE
- <timestamp>.dba Database startup or shutdown
- <timestamp>.rmv Change data file assignment of a tablespace
- <timestamp>.rli Reorganize a list of objects
- <timestamp>.rsn Restore or recovery to current time
- <timestamp>.rsp Restore or recovery to point-in-time
- <timestamp>.rst Restore or recover the database
- struct<DBSID>.log Structure log
- reorg<DBSID>.log Main log

For more information on SAPDBA logs, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

SAPDBA writes scripts, parameter files, and restart files as follows to the directory sapreorg/<datestamp>/:

- alt<TSP>.sql Delete index analysis
- con<TSP>.sql Create constraints
- cssvg.sql Create constraints, grants, synonyms, sequences, and views
- dro<TSP>.sql Drop table or tablespace
- grants.sql Create grants
- ind<TSP>.sql Create indexes
- inx<TSP>.sql Test script
- swi<TSP>.sql Switch redo log file, backup control file
- tsp<TSP>.sql Create tablespace PSAP<TSP>
- tab<TSP>.sql Create tables
- del<TSP>.sh Delete various objects
- exp<TSP>.sh Export tables and data
- imp<TSP>.sh Import tables and data
- exc<TSP>.sh Compress or split export dump files
- imc<TSP>.sh Decompress or merge export dump files for import
- exp<TSP>.par Parameter file for export
- exc<TSP>.par Parameter file for splitting export dump files
- restart.rli Restart for reorganization list
- restart.rmv Restart for move or rename
- restart.rsi Restart for reorganization of table or index
- restart.rtc Restart for reorganization of tablespace
- restart.rtd Restart for reorganization of tablespace with data file

13.12 Directory Structure (UNIX)

- `restart.exs` Restart for export or import of tables and index definitions
- `restart.exd` Restart for export or import of tables with indexes and data

For more information on files with the endings:

`sh` or `sql`, see [SAPDBA Scripts and Files \[Page 214\]](#)

`par` see [SAPDBA Log Files and Their Contents \[Page 205\]](#)

`rsi`, `rli`, `rtc`, `rtd`, or `rmv`, see [Restart File for Reorganizations \[Page 279\]](#)

`exs` or `exd`, see [Restart File for Export or Import \[Page 302\]](#)

- `sapcheck/`
 - `<timestamp>.chk` Log of `sapdba -check`
 - `<timestamp>.nxt` Log of `sapdba -next`
 - `<timestamp>.aly` Log of `sapdba -analyze`
 - `<timestamp>.opt` Log of `sapdba -checkopt`
 - `<timestamp>.sta` Log of `sapdba -statistics`

For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

- `sapbackup/`
 - `back<DBSID>.log` BRBACKUP summary log
 - `<coded timestamp>.xyz` BRBACKUP detail log, where:
 - `x = a` (whole, previously **all**), `p` (**partial**), `f` (**full**), `i` (**incremental**)
 - `y = n` (**online**) or `f` (**offline**)
 - `z = t` (**tape**), `p` (**pipe**), `d` (**disk**), `f` (**util_file**), `r` (**RMAN**), `s` (**remote disk, stage**),
 - `xyz = qub` Query for which tapes are to be used
 - `xyz = cmb` Determination of software compression rate
 - `xyz = tib` Tape initialization
 - `xyz = rmp` RMAN preparation run
 - `xyz = dbv` Database verification with DBVERIFY
 - `xyz = ddb` Delete of disk backup
 - `<coded timestamp>.xyz` BRRESTORE detail log, where:
 - `xyz = rsb` Restore backup files
 - `xyz = rsa` Restore archive files
 - `xyz = rsf` Restore individual files
 - `<coded timestamp>/` Disk backups
 - `cntrl.dbf` If online backup was performed
 - `ctrl<DBSID>.dbf` If offline backup was performed
 - `back<DBSID>.log` Summary log
 - `init<DBSID>.ora`

- `init<DBSID>.sap`
- `<coded timestamp>.xyz` Detail log
- `saparch/` Offline redo logs or archive logs
 - `<DBSID>arch<thread_number>_<log seq.number>.dbf`
 - For example: `C11arch1_4711.dbf` (Default is OS-dependent)
 - `arch<DBSID>.log` BRARCHIVE summary log
 - Detail logs:
 - `<coded timestamp>.sve` Original saved
 - `<coded timestamp>.svd` Original saved and deleted
 - `<coded timestamp>.cpy` Original copied
 - `<coded timestamp>.cpd` Original copied and deleted
 - `<coded timestamp>.dev` Deleted, were saved once
 - `<coded timestamp>.dcp` Deleted, were saved twice
 - `<coded timestamp>.ssv` Parallel saved on two stations
 - `<coded timestamp>.ssd` Parallel saved on two stations and deleted
 - `<coded timestamp>.cps` Copy and save
 - `<coded timestamp>.cds` Copy, delete and save
 - `<coded timestamp>.qua` Query which tapes to be used
 - `<coded timestamp>.cma` Determination of software compression rate
 - `<coded timestamp>.tia` Tape initialization
 - `<coded timestamp>.fst` Stop archiving using `brarchive -f stop`
- `saptrace/`
 - `background/`
 - `alert_<DBSID>.log` Oracle Alert file
 - `*.trc` Oracle trace files
 - `usertrace/`
 - `*.trc` User trace files

Executables in Directory `/usr/sap/<SAPSID>/SYS/exe/run`

- `sapdba`
- `brbackup`
- `brarchive`
- `brconnect`
- `brrestore`
- `brtools`

13.13 Directory Structure (Windows NT)

- ...

13.13 Directory Structure (Windows NT)

Definition

This section describes the directory structure for the Oracle database with the UNIX operating system.

Use

The directories contain a range of files such as profiles, log files, scripts, executables, and so on.

The following conventions apply to this section:

Term	Meaning
%<name>%	Environment variable
<DBSID>	;%ORACLE_DBSID%
<TSP>	Tablespace short name For example, BTABD for the tablespace PSAPBTABD.

Structure

There are the following main structures:

- %ORACLE_HOME% for Oracle-specific objects with the default directory
 \oracle\<DBSID>
- %SAPDATA_HOME% for SAP-specific objects with the default directory
 \oracle\<DBSID>
- Executables with the directory x:\usr\sap\<DBSID>\SYS\exe\run

Oracle Home Directory

This directory contains the following subdirectories:

- database\
 - init<DBSID>.ora Profile for Oracle
 - init<DBSID>.dba Profile for SAPDBA
 - init<DBSID>.sap Profile for BRBACKUP, BRARCHIVE, and BRRESTORE
 - cntrl<DBSID>.dbf Control file
- rdbms\
- bin\

Contains binaries such as svrmgrl, sqlplus, exp, imp, and so on

SAP Home Directory

For the files starting log_, g means group and m means member:

- origlogA\
 - log_tg101m1.dbf

13.13 Directory Structure (Windows NT)

- log_tg103m1.dbf
- origlogB\
 - log_tg102m1.dbf
 - log_tg104m1.dbf

Mirrored redo logs are optional – although we strongly recommend using them – so the specification for mirrlogA and mirrlogB might vary:

- mirrlogA\
 - log_tg101m2.dbf
 - log_tg103m2.dbf
- mirrlogB\
 - log_tg102m2.dbf
 - log_tg104m2.dbf
- sapdata1\
 - system_1\system.data1
 - btabd_1\btabd.data1 Example of a SAP data file for tablespace PSAPBTABD
 - ...
- sapdata2\
 - btabi_1\btabi.data1 Example of a SAP data file for tablespace PSAPBTABI
- sapdata<n>\
- sapreorg\

Logs written by SAPDBA:

- <timestamp>.rch Log check option
- <timestamp>.ral Alter storage in reorganization menu
- <timestamp>.rsi Reorganize a single table or index
- <timestamp>.rtc Reorganize tablespace without data file
- <timestamp>.rtd Reorganize tablespace with data file
- <timestamp>.ext Tablespace extension
- <timestamp>.exs Export table and index structures
- <timestamp>.exd Export table with data
- <timestamp>.exg Export constraints, synonyms, sequences, views, grants
- <timestamp>.ims Import structure
- <timestamp>.imd Import table with data
- <timestamp>.rcv Recovery
- <timestamp>.brb Backup start with BRBACKUP
- <timestamp>.bra Archive start with BRARCHIVE

13.13 Directory Structure (Windows NT)

- <timestamp>.dba Database startup or shutdown
- <timestamp>.rmv Change data file assignment of a tablespace
- <timestamp>.rli Reorganize a list of objects
- <timestamp>.rsn Restore or recovery to current time
- <timestamp>.rsp Restore or recovery to point-in-time
- <timestamp>.rst Restore or recover the database
- struct<SID>.log Structure log
- reorg<SID>.log Main log

For more information on SAPDBA logs, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

SAPDBA writes scripts, parameter files, and restart files as follows to the directory sapreorg\<datestamp>\:

- alt<TSP>.sql Delete index analysis
- con<TSP>.sql Create constraints
- cssvg.sql Create constraints, grants, synonyms, sequences, and views
- dro<TSP>.sql Drop table or tablespace
- grants.sql Create grants
- ind<TSP>.sql Create indexes
- inx<TSP>.sql Test script
- swi<TSP>.sql Switch redo log file, backup control file
- tsp<TSP>.sql Create tablespace PSAP<TSP>
- tab<TSP>.sql Create tables
- del<TSP>.sh Delete various objects
- exp<TSP>.sh Export tables and data
- imp<TSP>.sh Import tables and data
- exc<TSP>.sh Compress or split export dump files
- imc<TSP>.sh Decompress or merge export dump files for import
- exp<TSP>.par Parameter file for export
- exc<TSP>.par Parameter file for splitting export dump files
- restart.rli Restart for reorganization list
- restart.rmv Restart for move or rename
- restart.rsi Restart for reorganization of table or index
- restart.rtc Restart for reorganization of tablespace
- restart.rtd Restart for reorganization of tablespace with data file
- restart.exs Restart for export or import of tables and index definitions

13.13 Directory Structure (Windows NT)

- `restart.exd` Restart for export or import of tables with indexes and data
- `exp<TSP>.dmp` Export dump file

For more information on files with the endings:

`sh` or `sql`, see [SAPDBA Scripts and Files \[Page 214\]](#)

`par` see [SAPDBA Log Files and Their Contents \[Page 205\]](#)

`rsi`, `rli`, `rtc`, `rtd`, or `rmv`, see [Restart File for Reorganizations \[Page 279\]](#)

`exs` or `exd`, see [Restart File for Export or Import \[Page 302\]](#)

- `sapcheck\`

`conn<DBSID>.log` BRCONNECT summary log

BRCONNECT detail logs:

- `<coded timestamp>.chk` Log of `-check` function
- `<coded timestamp>.nxt` Log of `-next` function
- `<coded timestamp>.aly` Log of `-analyze` function
- `<coded timestamp>.opt` Log of `-checkopt` function
- `<coded timestamp>.sta` Log of `-statistics` function
- `<coded timestamp>.cln` Log of `-cleanup` function

For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

- `sapbackup\`

- `back<DBSID>.log` BRBACKUP summary log

- `rest<DBSID>.log` BRRESTORE summary log

- `<coded timestamp>.xyz` BRBACKUP detail log, where:

- `x = a` (whole, previously **all**), `p` (**p**artial), `f` (**f**ull), `,`, `i` (**i**ncremental)
- `y = n` (online) or `f` (offline)
- `z = t` (**t**ape), `p` (**p**ipe), `d` (**d**isk), `f` (util_**f**ile), `r` (**R**MAN), `s` (remote disk, **s**tage)
- `xyz = qub` Query for which tapes are to be used
- `xyz = cmb` Determination of software compression rate
- `xyz = tib` Tape initialization
- `xyz = rmp` RMAN preparation run
- `xyz = dbv` Database verification with DBVERIFY
- `xyz = ddb` Delete of disk backup

- `<coded timestamp>.xyz` BRRESTORE detail log, where:

- `xyz = rsb` Restore backup files
- `xyz = rsa` Restore archive files
- `xyz = rsf` Restore individual files

- `<coded timestamp>\` Disk backups

13.13 Directory Structure (Windows NT)

- cntrl.dbf If online backup was performed
- ctrl<DBSID>.dbf If offline backup was performed
- back<DBSID>.log Summary log
- init<DBSID>.ora
- init<DBSID>.sap
- <coded timestamp>.xyz Detail log
- saparch\ Offline redo logs or archive logs
 - <DBSID>arch<thread_number>_<log seq.number>.dbf
 - For example: C11arch1_4711.dbf (Default is OS-dependent)
 - arch<DBSID>.log BRARCHIVE summary log
- BRARCHIVE detail logs:
 - <coded timestamp>.sve Original saved
 - <coded timestamp>.svd Original saved and deleted
 - <coded timestamp>.cpy Original copied
 - <coded timestamp>.cpd Original copied and deleted
 - <coded timestamp>.dev Deleted, were saved once
 - <coded timestamp>.dcp Deleted, were saved twice
 - <coded timestamp>.ssv Parallel saved on two stations
 - <coded timestamp>.ssd Parallel saved on two stations and deleted
 - <coded timestamp>.cps Copy and save
 - <coded timestamp>.cds Copy, delete and save
 - <coded timestamp>.qua Query which tapes to be used
 - <coded timestamp>.cma Determination of software compression rate
 - <coded timestamp>.tia Tape initialization
 - <coded timestamp>.fst Stop archiving using brarchive -f stop
- saptrace\
 - background\
 - alert_<DBSID>.log Oracle Alert file
 - *.trc Oracle trace files
 - usertrace\
 - *.trc User trace files

Executables in Directory <drive>:\orant

- sapdba
- brbackup

- brarchive
- brconnect
- brrestore
- brtools
- compress
- uncompress
- cpio
- mt
- dd



Approach to Oracle DBA

Purpose

This section helps you to work out an approach to database administration (DBA) with the Oracle database. For more information on how to approach Oracle backup, see [Backup, Restore, and Recovery \[Page 73\]](#).

For more information on how to perform Oracle DBA with the tools supplied by SAP, see [Tools for Oracle DBA \[Page 183\]](#).

Prerequisites

You have already [started Oracle DBA with the SAP System \[Page 23\]](#).

Process Flow

Result

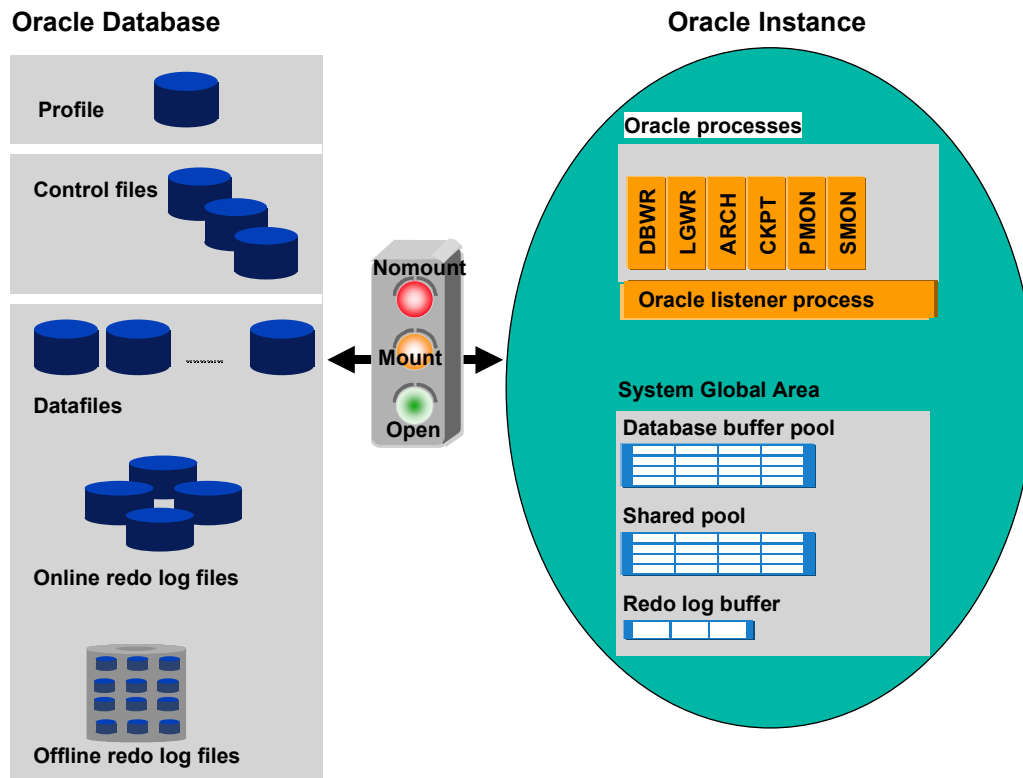
14 Database Mode

Purpose

This section tells you about setting the operating mode of the Oracle database.

Prerequisites

The main components of the database and instance are shown in the following graphic:



When an Oracle database is running, it is associated with an instance. The process of associating the database with an instance is called mounting the database. To make the mounted database accessible to authorized users, you must open it.

Process Flow



We **strongly recommend** you to use BRCONNECT functions to start and stop the database rather than the equivalent SAPDBA ones (if one is available, this is indicated below). The reason is that we are no longer developing the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Starting the Database

The following describes how you can start the database:

Type	What happens	How
No mount	Database instance is built up	SAPDBA <i>Startup NOMOUNT</i>

	Operating system resources are allocated using configuration information stored in the profile <code>init<SID>.ora</code> .	
Mount	Database control files are evaluated. Information about the file structure of the database is read Datafiles and logs are not yet opened.	SAPDBA <i>Startup MOUNT</i>
Open	All files in the database system are opened. If required, instance recovery is performed immediately after opening the database. Pending database transactions are ended.	<code>brconnect -f start</code> SAPDBA <i>Startup normal</i> SAPDBA <i>Startup RESTRICT</i> – this is restricted mode [Page 221] (privileged user access only) for database administration tasks such as reorganization [Page 259]

Stopping the Database

The following describes how you can stop the database:

Type	What happens	How
Normal	No new database logon possible. After all database user have logged off, the database is closed properly: all files are closed, the database is dismounted, and the instance is shut down. The database is consistent after shutdown.	<code>brconnect -f dbshut</code> SAPDBA <i>Shutdown normal</i>
Immediate	Only the current commands are executed. PMON ends all sessions and performs a rollback of the open transactions. The database is then closed properly (as for a normal shutdown). The database is consistent after shutdown. DBWR and ARCH might require up to 1 hour post-processing time.	SAPDBA <i>Shutdown immediate</i> SAPDBA <i>Shutdown immediate force</i> – this forcibly disconnects any SAP users that are still connected
Abort	Emergency database shutdown Users are not logged off and open transactions are not rolled back. The database is not consistent after shutdown. An instance recovery is automatically performed at the next database startup.	SAPDBA <i>Shutdown Abort</i>

See [Startup and Shutdown of the Oracle Database with SAPDBA \[Page 217\]](#).

15 Database System Check

Purpose

This section helps you develop an approach to checking the database system. By running regular checks of the database system, you can identify potential problems and take the necessary action. You can schedule regular checks in the Computing Center Management System (CCMS) of the SAP System, using the DBA Planning Calendar. You can also run checks with BRCONNECT.

Prerequisites

For more information on how BRCONNECT performs database system check, see [Database System Check with BRCONNECT \[Page 409\]](#). This is the same functionality as used in the DBA Planning Calendar.

Process Flow

1. You choose a method to run the database system check:
 - Regularly, preferably daily, by [scheduling it in the DBA Planning Calendar of CCMS \[Ext.\]](#) – recommended for normal use



We recommend you to use the [action patterns \[Ext.\]](#) in the DBA Planning Calendar. These enable you to schedule the database system check daily.

- One-off by using the BRCONNECT command [-f check \[Page 431\]](#) – recommended for non-standard checks



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -check \[Page 347\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

2. You check the results using CCMS and take any necessary action:
 - You normally use the [alert monitor \[Ext.\]](#) to check the results of the system check.
 - For a detailed technical view, see [Displaying Alert Messages from Database System Check \[Ext.\]](#) and [Parameters and Alerts for Database System Check \[Ext.\]](#).
3. If required, you configure database system check using CCMS. This includes activating or deactivating conditions and changing the threshold and severity levels (that is, error, warning, or exception). For more information, see [Configuring Database System Check \(Oracle\) \[Ext.\]](#).

16 Space Management

Purpose

This section helps you develop an approach to managing the space of your Oracle database.

Process Flow

1. You consider whether to use raw devices or a file system. Raw devices are generally 10 to 20% faster on UNIX systems. However, with [Veritas Quick I/O \[Page 179\]](#) the difference is reduced.

We recommend raw devices only for experienced database administrators because the administration is more complex. For example, only one Oracle file can be set up on each raw device.

2. You use the Oracle AUTOEXTEND option, recommended starting with Oracle Release 8. AUTOEXTEND means that tablespaces are **automatically** extended when they run out of space.

We recommend using this when you:

- [Create a new tablespace \[Page 235\]](#)
- [Extend an existing tablespace \[Page 233\]](#) by adding a new file
- [Resize the existing data files of a tablespace \[Page 275\]](#)

For more information on AUTOEXTEND, see the Oracle documentation.

3. You set up locally managed tablespaces when creating [new tablespaces \[Page 235\]](#) and during a [reorganization \[Page 242\]](#), except for system and rollback tablespaces.

For more information on our recommendations for locally managed tablespaces, see SAP Note 416772.

4. You [check the database system \[Page 69\]](#) regularly, so that you can identify potential problems in space management early.
5. If you identify a problem with the tablespaces, you perform [tablespace administration \[Page 229\]](#).
6. If you identify a problem that might require reorganization, you consider the following:
 - Indexes can often become unbalanced, requiring reorganization. However, this does **not** occur with locally managed tablespaces, because fragmentation is no longer possible.
 - Reorganization is necessary if you switch from normally (that is, dictionary-) managed to locally managed tablespaces.
 - Fragmentation is often a reason for reorganization:
 - Internal fragmentation: fragmentation of table or index data within a block or chaining of data over several blocks
 - External fragmentation: fragmentation of the extents of a logical object and of the free space within a tablespace, maximum number of extents for an object reached

For more information if you decide to reorganize, see [Reorganization \[Page 242\]](#).

17 Update Statistics

Purpose

By running update statistics regularly, you make sure that the database statistics are up-to-date, so improving database performance. You can schedule the checks in the Computing Center Management System (CCMS) of the SAP System, using the DBA Planning Calendar.

The Oracle cost-based optimizer (CBO) uses the statistics to optimize access paths when retrieving data for queries. If the statistics are out-of-date, the CBO might generate inappropriate access paths (such as using the wrong index), resulting in poor performance.

From Release 4.0, the CBO is a standard part of the SAP System. If statistics are available for a table, the database system uses the cost-based optimizer. Otherwise, it uses the rule-based optimizer.

Update statistics supports partitioned tables. For more information, see SAP Note 424243.

Prerequisites

For more information on how BRCONNECT performs update statistics, see [Update Statistics with BRCONNECT \[Page 420\]](#).

You can improve the performance of update statistics as follows:

- Parallel processing

You can implement this with BRCONNECT or DBMS_STATS:

- BRCONNECT

You can implement BRCONNECT parallel processing as follows:

- Command call `brconnect -p <number of threads>`
- Parameter `stats_parallel_degree` in the `init<DBSID>.sap` file

Here are some examples:



```
brconnect -c -u / -f stats -p 4
stats_parallel_degree = 4
```

Each of these sets the number of threads – that is, the degree of parallelism – to 4. The default is 1, which means that update statistics runs in non-parallel – that is, serial – mode. In either case, each table is processed by a single thread. For more information, see SAP Note 403713.

- DBMS_STATS Package

There is a parallel processing option in this package that considerably reduces run times for very large tables (that is, several hundred GB). Each table can be processed by several threads.

For more information, see SAP Note 408532.

You can use both the above types of parallel processing in the same BRCONNECT run.

- Oracle table monitoring

With this feature, the Oracle database system constantly updates information concerning record counts for database tables, entering the results in the `DBA_TAB_MODIFICATIONS`

table. It only takes BRCONNECT a very short time to read this results table and determine whether update statistics is required for a given database table.

For more information, see SAP Note 408527.

Process Flow

You choose one of the following approaches to update statistics:

- DBA Planning Calendar in the Computing Center Management System (CCMS)

For more information, see [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#). The DBA Planning Calendar uses the BRCONNECT commands.



We recommend you to use this approach in production operation because you can easily schedule update statistics to run **automatically** at regular intervals. We recommend you to update statistics weekly.

- [BRCONNECT \[Page 420\]](#)
- [SAPDBA \[Page 222\]](#) – **not** recommended



We **strongly recommend** you to use the [update statistics in BRCONNECT \[Page 420\]](#) rather than the equivalent SAPDBA functionality:

- [sapdba -statistics \[Page 353\]](#)
- [sapdba -analyze \[Page 345\]](#)
- [sapdba -delete \[Page 350\]](#)
- [sapdba -checkopt \[Page 348\]](#)

The reason is that we are no longer developing the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.



Backup, Restore, and Recovery

Use

You need to regularly back up your Oracle database so that, in the event of failure, you can restore and recover it.

Integration

You use the following tools for backup, restore, and recovery:

Tool	Use
DBA Planning Calendar [Ext.] – in the Computing Center Management System (CCMS) of the SAP System	Routine backups: <ul style="list-style-type: none"> • Backup of database files • Backup of offline redo log files
SAPDBA for Oracle [Page 184]	Backup of database files [Page 303] Backup of offline redo log files [Page 313] Restore and recovery [Page 321]
BRBACKUP [Page 360]	Backup of database files
BRARCHIVE [Page 383]	Backup of offline redo log files
BRRESTORE [Page 395]	Restore of database and offline redo log files
SQLPLUS [Page 487]	Database recovery when SAPDBA is not sufficient
Oracle Recovery Manager (RMAN) [Page 137] – integrated with BRBACKUP and BRARCHIVE	Backup, restore, and recovery

Prerequisites

You familiarize yourself with the above tools and make sure that you **frequently** back up the **entire** database. For more information, see [Backup Overview \[Page 75\]](#).

Features

- Backup
 - BRBACKUP backs up database files
 - BRARCHIVE backs up offline redo log files
- Restore

BRRESTORE restores data files that have been damaged or are missing, using backups of the database files.
- Recovery

SAPDBA recovers transactions lost since the database backup, using backups of the offline redo log files to roll forward the lost transactions.

For more information, see [Why Back Up the Database? \[Page 76\]](#) and [What Needs Backing Up? \[Page 77\]](#)

Activities

- Backup

We recommend you to use the DBA Planning Calendar for routine backup tasks, because this enables you to automatically schedule the backup.

- Restore and recovery

You can perform backup, restore, and recovery tasks from either of the following:

- The menus in SAPDBA

SAPDBA calls the tools BRBACKUP, BRARCHIVE, or BRRESTORE as necessary to complete the task you have chosen.

- The command line

In this way, you can use the tools BRBACKUP, BRARCHIVE, or BRRESTORE, but this requires expert knowledge.



We recommend you to use the DBA Planning Calendar for routine backup because this enables you to automatically schedule the backup.

We recommend you to use the SAPDBA menus for restore and recovery because SAPDBA guides you through the necessary steps.

18 Backup Overview

This section gives you basic information to develop a careful approach to backing up your Oracle database. For more information on the reasons for database backup, see [Why Back Up the Database? \[Page 76\]](#)

Prerequisites

When designing your approach to backup, archive, and recovery, consider the following:

- You design an approach based on the needs of your company. You ask yourself questions such as:
 - What level of availability do you require from the database?
 - How long can you afford to shut down the SAP System in the event of data loss? Some backup approaches require a longer restore and recovery time than others.
 - Can you afford to lose data at all? If not, consider [high availability solutions for Oracle \[Ext.\]](#) such as [Oracle standby databases \[Ext.\]](#).
- You carefully test your approach **before** your SAP System goes live, and again **after** any changes have been made to the approach.
- You document your approach in a plan and make sure that all relevant people know the procedures to follow in the event of problems.

Process Flow

1. You identify [what needs to be backed up \[Page 77\]](#).

Normally, you back up the complete database and the redo log files.

2. You identify the [database backup type \[Page 80\]](#) that you require.

Normally, you perform an [online \[Page 82\]](#) and [complete \[Page 83\]](#) backup.

We recommend performing at least one offline backup per cycle, more often (for example, weekly) if possible. Only use [incremental backup \[Page 83\]](#) for larger databases.

3. You work out a [backup cycle \[Page 84\]](#). We recommend a minimum cycle of 14 days, although 28 days is preferable. For example, with a 28-day cycle, you reuse the backup media after 28 days.
4. You schedule regular backups.



We strongly recommend you to schedule regular backups with the [DBA Planning Calendar \[Ext.\]](#), using the [action patterns \[Ext.\]](#) available there. There are action patterns for different requirements to cover the main aspects of database administration, including backup.

5. You [verify \[Page 118\]](#) the:

- Backup tape readability, that is, a check on the contents of the media after the backup
- Database block consistency, that is, a check on the database itself

If possible, run both types of verify daily, otherwise weekly. At the least, be sure to run a verify once in each backup cycle.

18.1 Why Back Up the Database?

You can back up the database and then verify both the backup media and the database using a single BRBACKUP command, [brbackup -w use dbv \[Page 376\]](#). This option is also available in the [action patterns \[Ext.\]](#) of the [DBA Planning Calendar \[Ext.\]](#). Be aware that a verify considerably extends backup run times.

To only verify database block consistency (that is, without a database backup), use the command [brbackup -w only dbv \[Page 376\]](#).

To verify the backed up redo log files, use the command [brarchive -w\]-verify \[Page 391\]](#).

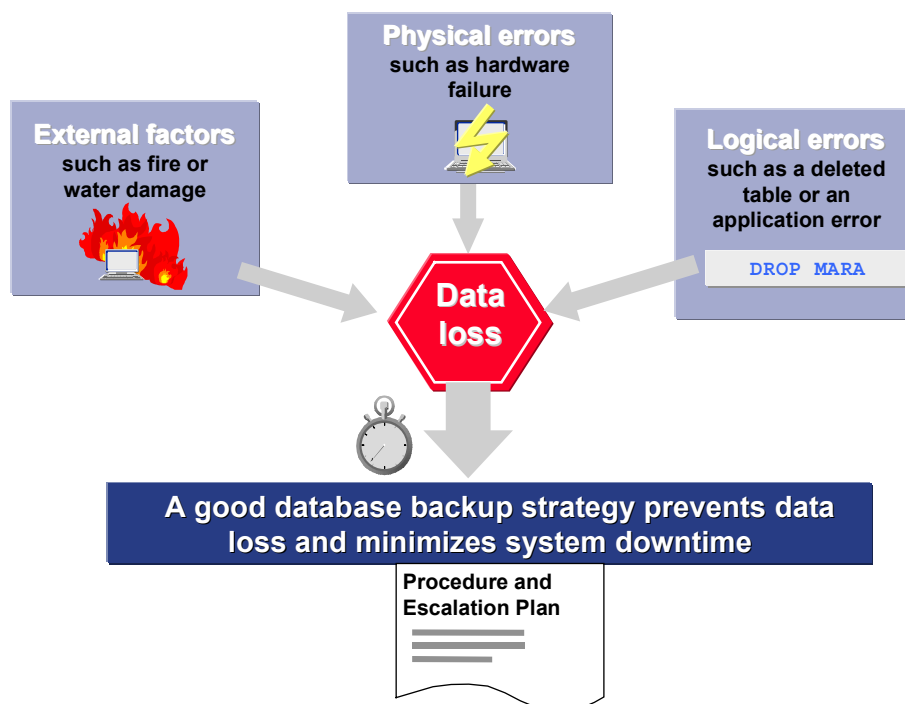
Result

See the following for examples of backup approaches:

- [Backup Approach with Daily Complete Backups \[Page 85\]](#)
- [Backup Approach for Very Large Database with Partial Backups \[Page 87\]](#)
- [Backup Approach with Retention Period Zero \[Page 88\]](#)

18.1 Why Back Up the Database?

Without a careful approach to backing up your Oracle database, you run the risk of experiencing excessive system downtime and possibly losing data. The following graphic shows the main ways that data loss occurs:



How you react to data loss depends on how it was caused:

- By external factors or physical errors

You must recover the database up to the point in time when the database crashed. If a full recovery is possible, only the data of uncommitted transactions before the error is lost.

- By logical errors

18.2 What Needs Backing Up?

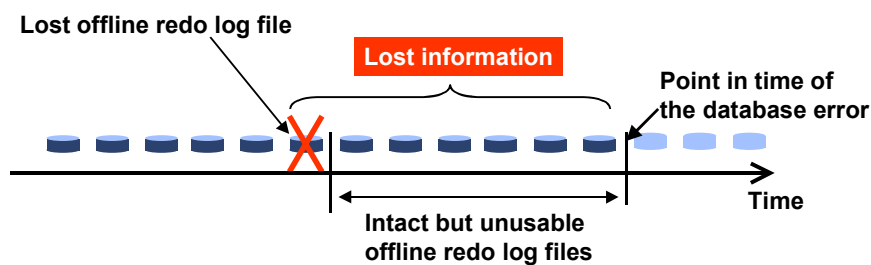
You must recover the database up to a point in time shortly before the error occurred. However, data entered **after** the error is lost.

To avoid data loss after a logical error, it is sometimes possible to restore the database to a different machine and then export the affected table from that machine to your production database. However, this method is difficult and requires expert knowledge of the application that uses the table.

As well as regularly backing up the database, you also need to archive the redo log files. The following graphic shows how important it is to archive the redo log files:

Forward recovery

A database backup is restored and you now want to recover data from offline redo log files



If one offline redo log file is lost, none of the files that follow it can be used

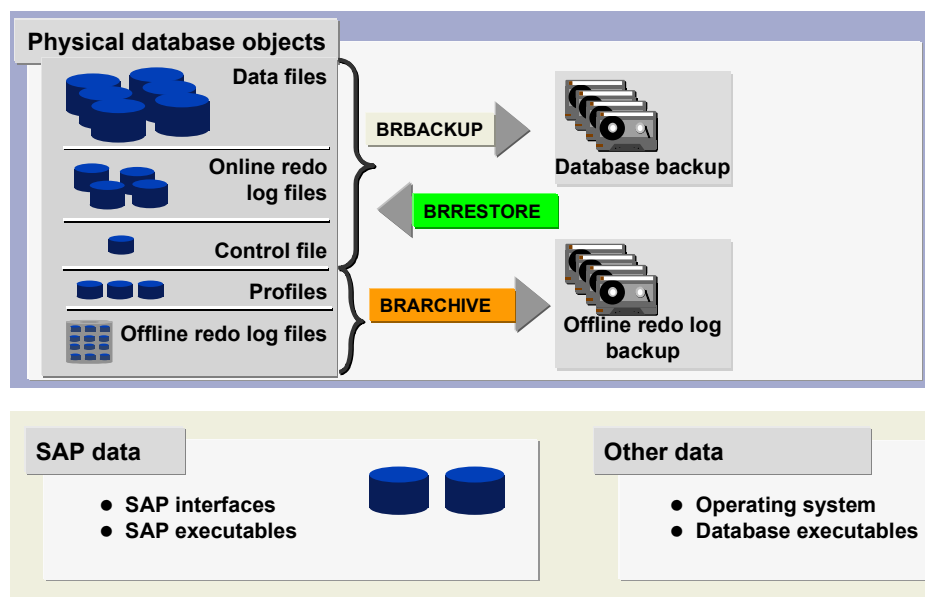


We recommend you to keep **at least two copies** of the offline redo log files on a secure storage medium. For maximum security, store the copies in different locations.

18.2 What Needs Backing Up?

Apart from deciding whether to perform an [online or offline \[Page 81\]](#), a [complete \[Page 83\]](#) or [incremental \[Page 83\]](#) backup of your Oracle database, you need to decide **what** to back up. The following graphic summarizes the different items that you need to consider in your backup approach:

18.2 What Needs Backing Up?



For more information on the tools, see:

- [BRBACKUP \[Page 360\]](#)
- [BRARCHIVE \[Page 383\]](#)
- [BRRESTORE \[Page 395\]](#)

The rest of this section discusses what you need to back up from a logical viewpoint.

Backing Up the Complete Database

- Consider the following when you decide the frequency of complete database backups:
 - The frequency of complete database backups should depend on the degree of activity in your database. High database activity increases the number of redo log files written between complete backups, which increases the time required for any necessary recovery.
 - Performing frequent complete backups reduces the number of redo log files that must exist in order to make a complete recovery. This reduces the data loss if one of these files is lost.

For more information, see [Backup Cycles \[Page 84\]](#).

- If a redo log file is lost, you can often not completely recover the database after an error, even if you have a complete database backup. Instead, you can only recover up to the gap in the redo log file sequence.
- 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 (that is, new, changed, or deleted tablespaces, or new datafiles). After a reorganization with datafiles, always back up the affected tablespace if you want to use the SAPDBA recovery functions. Follow the instructions for the tablespace backup below.
- Use BRBACKUP to back up the database and BRARCHIVE to archive the offline redo log files. See [Common Features of BRBACKUP and BRARCHIVE \[Page 357\]](#).

18.2 What Needs Backing Up?

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 have to be processed in order to update the tablespace. If you can back up the entire database on a daily basis, tablespace backups are not necessary.

However, tablespace backups are no replacement for frequent backups of the complete database because:

- 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, you decide what has to be backed up. BRBACKUP supports the backup operation itself, but does not help you decide which tablespaces to back up. Therefore, it is possible that you might forget to back up certain tablespaces. However, for operations such as a [tablespace extension \[Page 233\]](#) or a [reorganization \[Page 242\]](#), SAPDBA recommends you to back up the tablespace immediately.

You can use [tablespace backup \[Page 160\]](#) for large databases.

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. Therefore, you should 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 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 or 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.

Backing Up Executable Programs and Other SAP Components

In addition to backing up database files and offline redo log files, we recommend you to also back up the following **non-database** files:

- Permanent Files

You can find these files, for example, on UNIX systems in the subdirectories `/usr/sap/<SAPSID>/SYS` and `<ORACLE_HOME>/dbs` (UNIX) or `\\sapmnt\<SAPSID>\SYS` and `<ORACLE_HOME>` (Windows). They include executable programs and profiles of the SAP System and of the Oracle database system. We recommend you to back up the SAP directories after an SAP System upgrade and the Oracle directories after a database upgrade.

- Temporary Files

18.3 Database Backup Types

You can find these files, for example, on UNIX systems in the subdirectory `/usr/sap/<SAPSID>/<INSTANCE>`. The loss of these files is not critical, and does not cause data inconsistency. SAP provides tools that can reset the references to these files in the database, when required.

BRBACKUP backs up non-database files as well as database files, but we recommend you to only use it for this in exceptional circumstances. For more information, see [Backing Up Non-Database Files and Directories \[Page 361\]](#).



Backing up non-database files using BRBACKUP is **not** a replacement for backing up the file system at operating system level. For more information, see the documentation for your operating system.

18.3 Database Backup Types

Definition

This section describes the different types of backup that you can make of your Oracle database.



The term "backup" normally refers to a physical backup, which is sometimes called an image backup.

Oracle and SAP support the following types of database backup:

- Physical backup using [BRBACKUP \[Page 360\]](#)

Physical backups, sometimes called image backups, are complete block-for-block copies of the database, either [online or offline \[Page 81\]](#). Physical backups are required to recover the database to a consistent and current state. You can perform the following kinds of physical backups:

- [Consistent online backups \[Page 82\]](#), that is, while the database is in production use
- [Complete backups \[Page 83\]](#) of the entire database
- [Incremental backups \[Page 83\]](#) of only data that was changed since the last complete backup
- [Tablespace backups \[Page 160\]](#) to separately back up intensively used tablespaces between full backups – this is an advanced function to speed up recovery, which we do **not** normally recommend.
- [Partial backups \[Page 159\]](#) to back up only parts of the database, not the entire database.

- Logical backup using [SAPDBA \[Page 285\]](#) – also called export

SAPDBA export uses the Oracle export and import functions to let you back up and recover specific objects – that is, a set of database records – in the database. However, logical backups are **not** a suitable replacement for physical database backups because you can only recover the database to the condition at the time of the backup. You cannot recover database changes made after a logical backup.

SAP tables are usually used by multiple users or applications, which means that it is **not** a good idea to make user-related backups with Export/Import. For example, you do not want a user to restore the ATAB table in order to retrieve lost entries in a particular SAP pool table stored in ATAB.

18.3 Database Backup Types

Logical backups of SAP objects can be performed using the SAP tool *R3trans*. *R3trans* exports SAP 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 the documentation on *Transport Tools*.



There are also the following types of **non-database** backup:

- Backup of executable programs and other components of the SAP System, described in [What Needs Backing Up? \[Page 77\]](#)
- Operating system backup – for more information, see the documentation for your operating system

18.3.1 Online and Offline Backup

Definition

Online Backup

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.

Tablespace online backups on their own are inconsistent. To make the database consistent, you need to apply redo log entries from the backup period. If you use [RMAN \[Page 137\]](#) for online backup, it takes care of internal block consistency during the backup.

An online backup is made using operating system tools such as `cpio`, `tar`, or `dd` – for example, under the control of [BRBACKUP \[Page 360\]](#). Since these tools are not part of the database system, Oracle 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 performed in the event of an error. This process works as follows:

1. 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, that is, the system change number (SCN). When the next redo log file switch or checkpoint occurs (normal database operations continue), the header information remains unchanged.
2. Based on this mechanism, all the files of a tablespace are copied with uniform header information specifying when the backup was started.
3. Once the backup of the tablespace is complete, the command `ALTER TABLESPACE <tablespace name> END BACKUP` makes sure that the header information of the files is updated.

RMAN normally takes care of this process internally, so it is not explicitly performed.

Offline Backup

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 no longer up-to-date. In this case, you have to recover the database after you restore the backup, using the redo log files.

You must close the database for an offline backup, which means that you have to stop work in the SAP System. However, the SAP 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 SAP System is remade after the database is restarted. This means that the information in the buffers of the SAP System is not lost, which implies better performance immediately following the start.

18.3 Database Backup Types

18.3.2  Consistent Online Backup

Definition

A consistent online backup of your Oracle database is an alternative to an offline backup when you cannot close the database. An online backup has logically consistent data because the offline redo log files created during a backup are backed up with the database files on the same volume. You use [BRBACKUP \[Page 360\]](#) for a consistent online backup.

A consistent online backup differs from an offline backup in that a recovery of the database – that is, applying the redo log files – is always necessary in order to guarantee consistent data.

Distinguish between the following, which are completely independent:

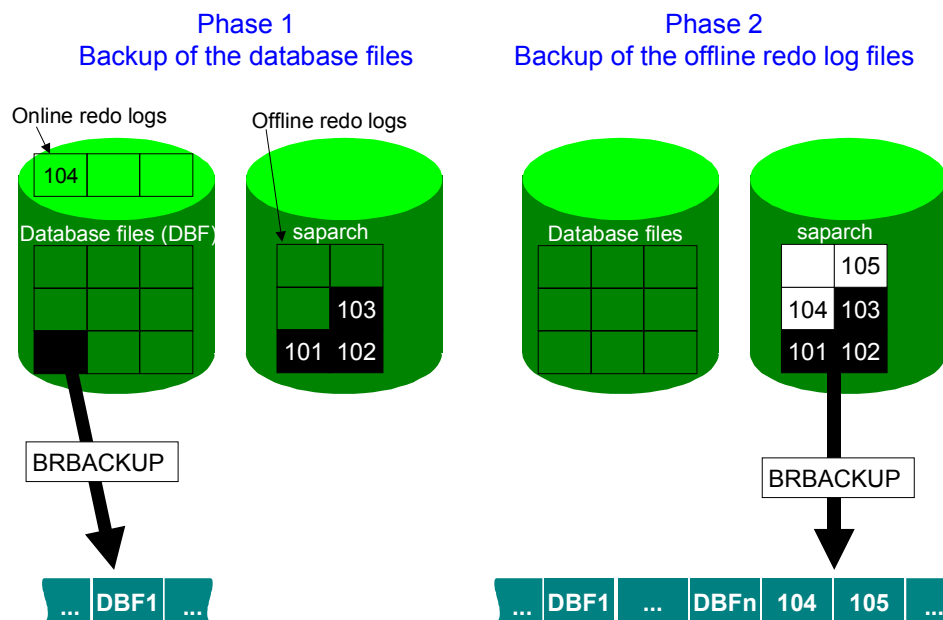
- Backup of the offline redo log files using BRBACKUP during a consistent online backup, as described in this section
- Backup of the offline redo log files using BRARCHIVE as a separate operation – for more information, see [-a|-archive \[Page 385\]](#).

Use

Database recovery is simplified because the offline redo log files of the same backup can be applied. In the same way the database can be reset to an earlier status (*Reset database*).



We recommend consistent online backups, especially for monthly and yearly backups, when an offline backup is unacceptable. However, consistent online backups **cannot** replace backups of the offline redo log files with BRARCHIVE.



To perform a consistent online backup, use the BRBACKUP option [-t|-type \[Page 374\]](#) `online_cons` or the relevant `init<DBSID>.sap` profile parameter [backup_type \[Page 450\]](#) = `online_cons`.

To restore the offline redo log files from the BRBACKUP backups, use the BRRESTORE option [-m|-mode \[Page 403\]](#) `archive_logs`. Restore a complete BRBACKUP backup including offline redo log files using the BRRESTORE option `-m full`.

18.3.3 Complete Backup

Definition

This section describes the different types of complete backup for the Oracle database.

SAP backup tools are integrated with the Oracle [Recovery Manager \(RMAN\) \[Page 137\]](#). You can use RMAN to make [incremental backups \[Page 83\]](#). However, you cannot start an incremental backup without a preceding full backup.

Complete backup refers to one of the following:

- Whole backup

Backs up all database files, but this backup is **not** cataloged, which means that you **cannot** use it as a reference backup for an incremental backup with RMAN. The syntax for a whole backup is as follows:

- In profile `init<DBSID>.sap:backup_mode = all`
- With BRBACKUP: `brbackup -m all`

- Full backup

Backs up all database files. You can perform 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. The syntax for a full backup is:

- In profile `init<DBSID>.sap:backup_mode = full`
- With BRBACKUP: `brbackup -m full`

In the context of the Recovery Manager this backup is a level-0 backup.

See also:

[backup_mode \[Page 449\]](#)

[-m|-mode \[Page 370\]](#)

[RMAN Backup Strategies \[Page 138\]](#)

18.3.4 Incremental Backup

Definition

This section describes incremental backups for the Oracle database. SAP backup tools support incremental backup with the [Oracle Recovery Manager \[Page 137\]](#) (RMAN). In an incremental backup, only the changes that have been made since the last [full \[Page 83\]](#) backup are saved.

Incremental backups improve the performance of the database backup, saving both time and memory. If you do not save as much time as expected, this is because each block has to be checked individually to see if it needs to be backed up, that is, whether it has been changed or not.

Incremental backups are especially useful for regular backups of [large databases \[Page 153\]](#).

18.4 Backup Cycles



An incremental backup **cannot** be used on its own to recover the database. You must always have the preceding full backup as well.

You can only make incremental backups with the RMAN. For more information, see [RMAN Backup Strategies \[Page 138\]](#).

Use

To be able to make an incremental backup, you must first make a full backup (level 0). A full backup of the database backs up all Oracle database blocks that have already been used.

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

Example

The following describes a weekly backup scenario:

- Sunday: 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 it comprises a backup (level 0) of the new files that were created as a result of tablespace extensions after the last full database backup. This means that you do **not** need to make a full backup of the entire database immediately after such extensions.

18.4 Backup Cycles

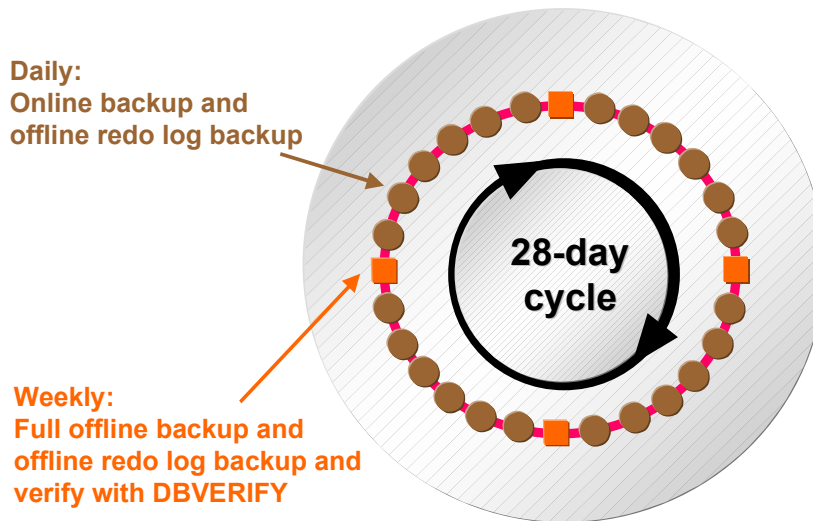
Definition

This section gives you recommendations on how to plan a backup cycle for your Oracle database. We recommend a backup cycle of at least 14 days, preferably 28 days. For more information on the tapes required, see [Backup Media \[Page 90\]](#).

Use

Recommended 28-Day Backup Cycle

18.5 Backup Approach with Daily Complete Backups



The guidelines are as follows:

- Perform a full **online** backup each working day.
- Perform a full **offline** backup at least once in the cycle, or at least weekly.
- Back up the offline redo log files each working day and after every online and offline backup. Be sure to back up the offline redo log files twice on separate tapes.
- To verify the process, you need to:
 - Verify backups for physical errors
 - Verify the database for logical errors at least once in the cycle
- Keep the verified full offline backup from each cycle in long-term storage, replacing it with a new initialized tape in the pool.
- Remember to perform additional backups after changes to the database structure and keep these tapes in long-term storage. You need to do this after any of the following:
 - A data file is added
 - A data file is moved to a different location
 - A tablespace or its data files are reorganized

18.5 Backup Approach with Daily Complete Backups

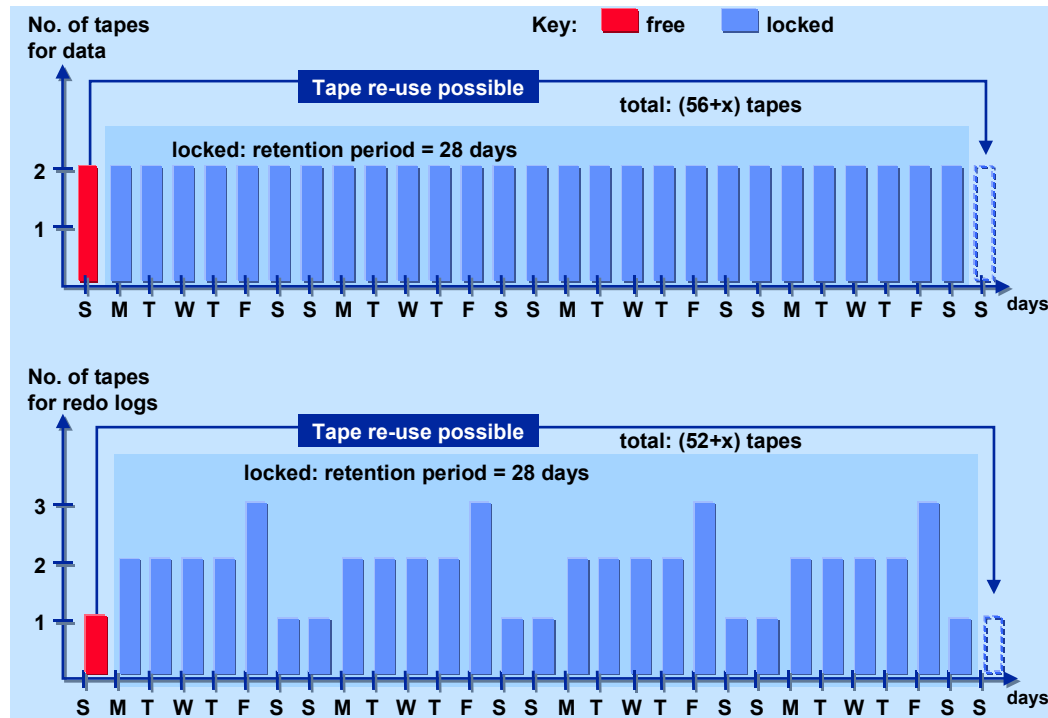
This section describes a sample backup approach for your Oracle database using daily [complete backups \[Page 83\]](#). This approach is secure and is suitable for small to medium-sized databases.

In this example, the size of the database is less than 200 GB, and daily backups are possible. As the SAP System does not have to be available after 18:00, the backup can be performed offline. Alternatively, the backup can be performed online at a time when the transaction load

18.5 Backup Approach with Daily Complete Backups

is low, such as overnight. If DLT drives are used, a full backup of the data (without redo information) fits onto two tapes.

Backup Approach with Daily Complete Backups and 28-Day Retention Period



To be able to deal with a faulty backup, several generations of backups must be available. In this example, the retention period is set to 28 days, so 27 backup generations are always available. The tape pool must also contain several reserve tapes. In this case, 56 + x tapes are required for data backup. The additional x tapes – approximately 20% of the required number – function as a reserve in case the amount of data to be backed up greatly increases, an extra unplanned backup becomes necessary, or a tape fails.

The redo information generated during the day – buffered on a separate disk that is as large as possible – is also backed up every day using a separate tape pool. As this data is necessary to recover a database after restoring a data backup, the retention period for the tapes must be no less than the retention period for the actual data backup. Particularly in the case of an online backup, you must always back up redo information **directly after** the data backup.



Without backups of the redo logs, the online backup is worthless.

As the redo data is much more dynamic than the actual data, even more reserve tapes are required. For this example, 52+x tapes are needed, where x is the number of reserve tapes for redo data. For security reasons, we recommend you to back up redo information **twice**, so the total number of tapes required is 2 x (52 + x). The actual number of tapes depends on the hardware implemented and the tape capacity available:

Capacity and Performance of Tapes and Tape Devices

Tape or Tape Device	Capacity (GB)	Approximate Rate (GB/hour)
IBM 3590/Magstar	20 – 40	10 – 15

18.6 Backup Approach for Very Large Database with Partial Backups

DLT 7000	35 – 70	15 – 20
DAT (DDS-3)	10 – 20	2 – 4
DST 310/312	50	30 – 50

See also:

[Backup Approach for Very Large Database with Partial Backups \[Page 87\]](#)

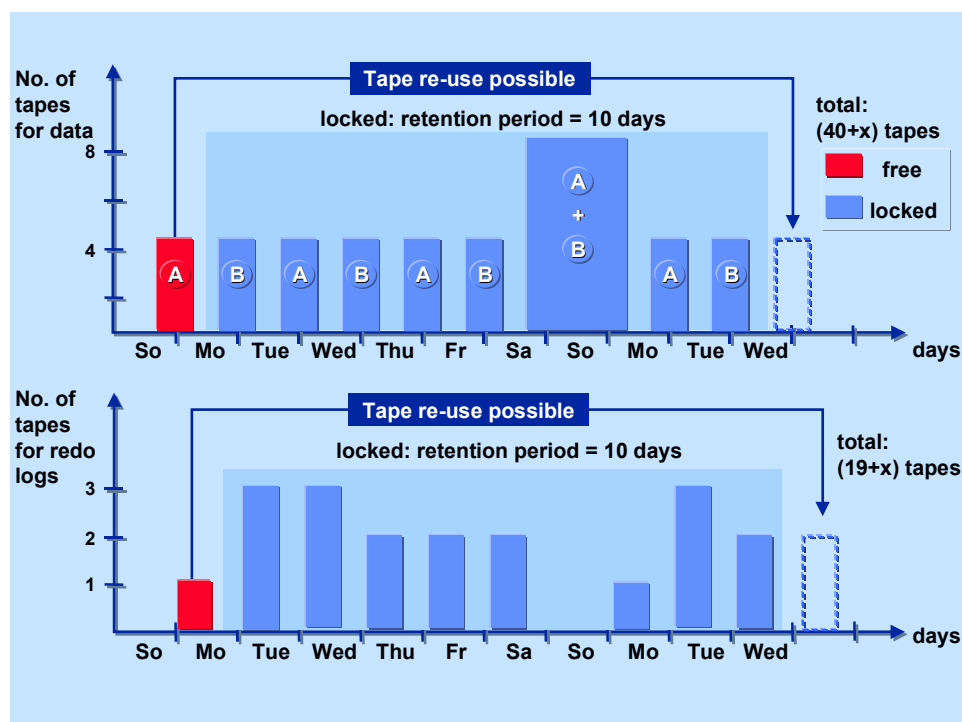
[Backup Approach with Retention Period Zero \[Page 88\]](#)

18.6 Backup Approach for Very Large Database with Partial Backups

This section describes a sample backup approach for a very large Oracle database using daily partial backups.

In this example, the database is too large for a complete daily offline or online backup and has to be available 24 hours a day on five working days. Therefore, the backup is spread over two days (part A and part B) and performed online. It runs during the night, as this is the only time when a low transaction load can be expected.

Backup Approach for a Very Large Database with Partial Backups A and B



This strategy is more error-prone than [the first example \[Page 85\]](#), because the database administrator is responsible for the correct distribution of the data to the partial backups, A and B. The risk of losing data is even higher, because online backups are only consistent in combination with redo information.

The number of tapes required for the data backup is $40+x$, where x is the number of reserve tapes. For security reasons, an additional full offline backup is performed at the weekend. If this strategy is used with a retention period of seven days, only four generations of backups

18.7 Backup Approach with Retention Period Zero

are available. The redo log files are even more important than in the first example, as the online backups are worthless without them. It is **essential** to back up the redo data twice. Therefore, you need $2 \times (19 + x)$ tapes for redo information, where x is the number of reserve tapes.



It is possible with [brarchive -cfs \[Page 390\]](#) to automatically create two copies of backup data using a single set of tapes.

See also:

[Backup Approach with Daily Complete Backups \[Page 85\]](#)

[Backup Approach with Retention Period Zero \[Page 88\]](#)

18.7 Backup Approach with Retention Period Zero

This section describes a sample backup approach for your Oracle database using daily full backups but with a retention period of zero.

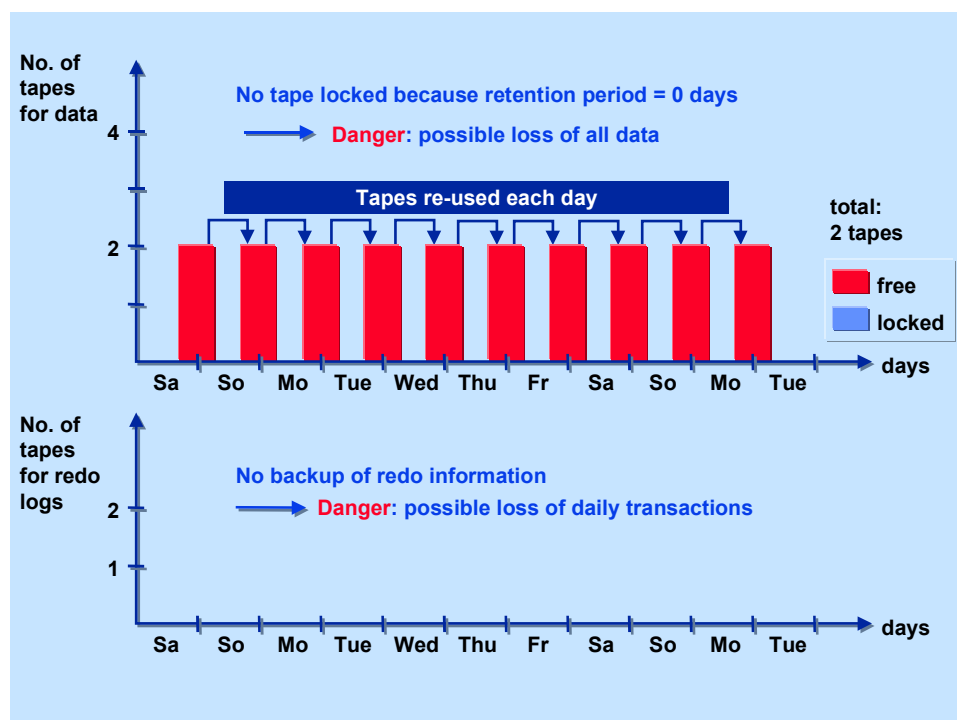


Do **not** follow the example shown in this section. It is included to illustrate a faulty backup approach.

In this example, a full backup is performed offline once a day. However, the retention period is set to zero days, so the two tapes required are overwritten each day.

In the event of a disk error, the single backup has to be used. If it cannot be read, the database is destroyed. As the redo information has not been saved separately, all transactions executed since the last backup are lost, if a disk failure affects data and redo information.

Faulty Backup Approach



See also:

18.7 Backup Approach with Retention Period Zero

[Backup Approach with Daily Complete Backups \[Page 85\]](#)

[Backup Approach for Very Large Database with Partial Backups \[Page 87\]](#)

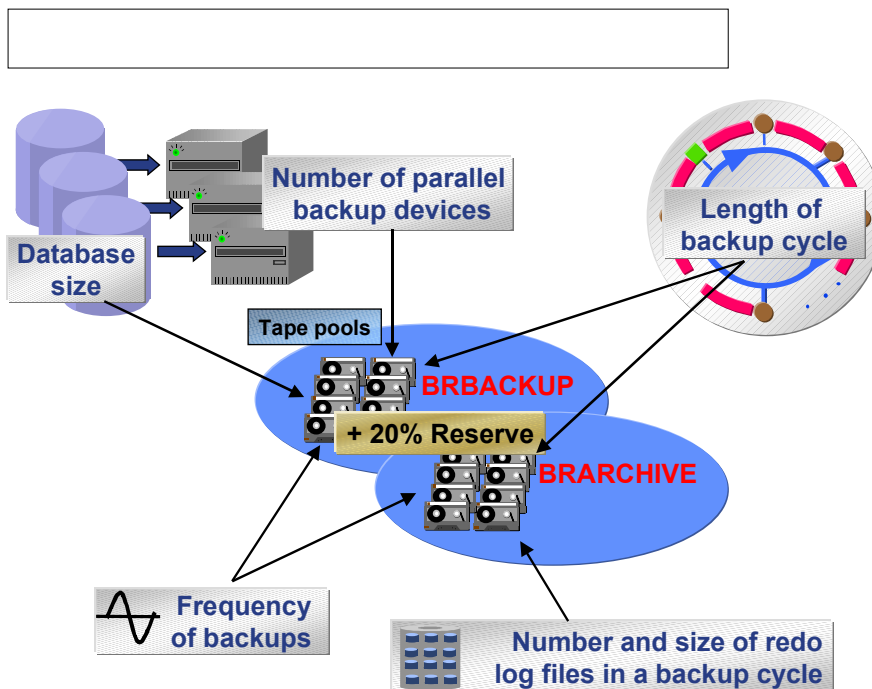
19 Backup Media

Definition

You require a pool of tapes for database and offline redo log file backups to back up your Oracle database. Ensure that enough tapes are provided in each tape pool to cover the entire backup cycle. We recommend having 20% more tapes than required to cover database growth and additional backups, for example after a new data file. Backup tapes can be reused at the end of a backup cycle (that is, normally after 28 days).

Of course, you can also back up the database to disk if you have enough storage space available and later copy it to tape.

The following graphic shows the factors you need to consider when making up a tape pool:



Backup media are normally **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 might want to back up the test database to backup devices that are connected to the host on which the production database is running.

For more information on how to manage the tape volumes, see [Volume Management \[Page 91\]](#).

For more information on data compression, see:

- [Software Compression \[Page 102\]](#)
- [Hardware Compression \[Page 101\]](#)

Integration

If you use an external backup, [BRBACKUP \[Page 360\]](#) calls BACKINT to administer backup media. For more information, see [External Backup Programs \[Page 180\]](#).

19.1 Volume Management

Use

The tape volumes for the Oracle database are overwritten again at each backup or archive by BRBACKUP or BRARCHIVE. These tools never use the space remaining on the tapes after a backup. New tapes have to be inserted each time you make a backup. For more information on the contents of a volume that has been written by BRBACKUP or BRARCHIVE, see [Used Volumes \[Page 96\]](#).

BRBACKUP or BRARCHIVE can only use volumes that are correctly [initialized \[Page 91\]](#). Initialized volumes have an SAP-specific [tape label \[Page 93\]](#).

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. For more information, see [Volume Expiration Period \[Page 95\]](#).

Activities

You can choose volumes as follows:

- [Select volumes automatically \[Page 100\]](#)
- [Select volumes manually \[Page 98\]](#)
- [Select volumes using external tools \[Page 99\]](#)

19.1.1 Volume Initialization

Use

You need to initialize the tape volumes for Oracle database backup with SAP tools. [BRBACKUP \[Page 360\]](#) or [BRARCHIVE \[Page 383\]](#) writes an SAP-specific label – that is, a file with the name `.tape.hdr0` – to the volume concerned. This label file is read when the volume is checked. If the label file does not exist, the check fails and the volume is rejected.

The following graphic summarizes volume initialization:

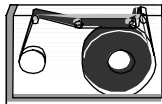
19.1 Volume Management

■ Profile *init<SID>.sap* contains the tape names:

```
...
volume_backup = (<SID>B01,<SID>B02...
volume_archive = (<SID>A01,<SID>A02...
...
```

■ Write the label to the tape that also contains the tape name

.tape.hdr0



■ Initialize new tapes, non-SAP tapes, or locked tapes:

brbackup -i force or *brarchive -i force*

■ Rename non-locked tapes:

brbackup -i -v <tape name> or *brarchive -i -v <tape name>*

Features

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 *tape_address_rew_arch*) are used serially during a volume initialization. For more information, see the parameters [tape_address \[Page 475\]](#) and [tape_address_rew \[Page 477\]](#). The volumes of all the available backup devices can be changed at the same time.



The information in the label is overwritten and the entire tape contents lost if you write to a BRBACKUP or BRARCHIVE volume directly using other tools.

Activities



You only have to initialize the following:

- New tapes
- Tapes that have **never** been used before by BRBACKUP or BRARCHIVE.

Only initialize the volumes once, not repeatedly before every backup. However, to change the name of a volume, you have to reinitialize it. You cannot change volume names during a backup.

1. You initialize all tape volumes to be used by BRBACKUP or BRARCHIVE for the first time. If you want to use [automatic tape management \[Page 100\]](#), the volume names must be listed in the *init<DBSID>.sap* parameters [volume_backup \[Page 481\]](#) or [volume_archive \[Page 480\]](#).



We recommend you to write paper labels on the volumes so that you can identify them more easily.

2. You enter the following options during an initialization:

19.1 Volume Management

- **-i | -initialize**

Renames volumes that have already been initialized. Only possible for volumes with an expired expiration period.

- **-i | -initialize force**

Initializes new volumes or volumes not yet used by BRBACKUP or 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 or 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**

Enters the names of volumes for initialization.

- **-n | -number**

Specified number of volumes for initialization.

The following examples show how you can use the above options:



Initialization of volumes with the volume names specified in *volume_backup/volume_archive*:
Use the BRBACKUP or 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* are initialized in sequence.
New and non-BRBACKUP or 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



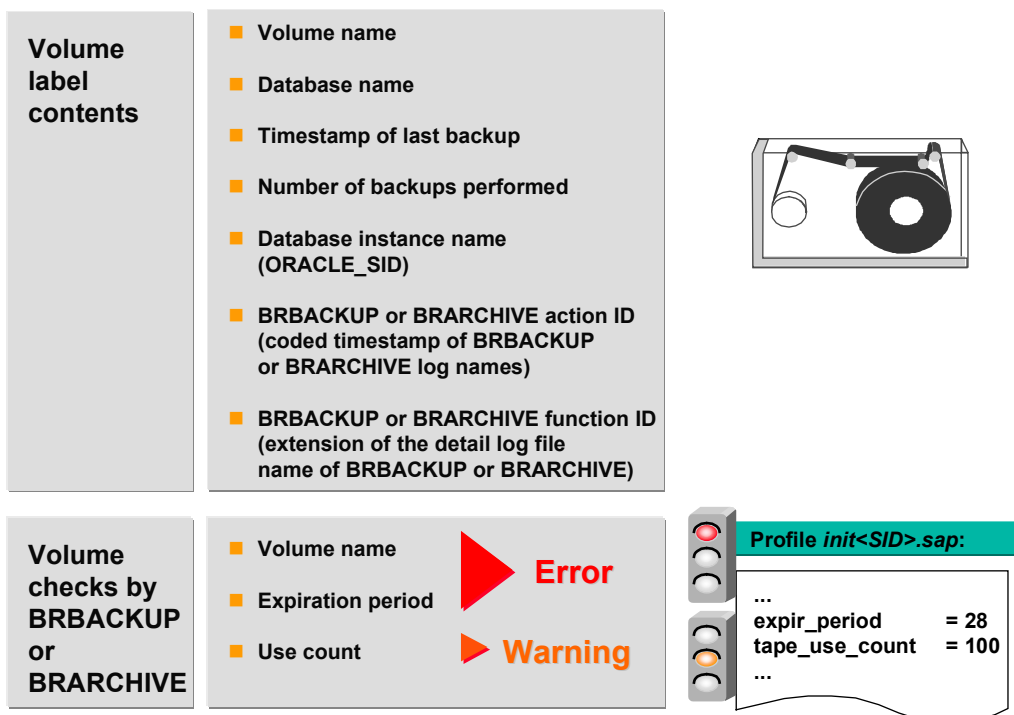
Initialization of volumes with a specific name:
brbackup|brarchive -i [force] -v <name1>[,<name2>]...
All the volumes for initialization must be mounted in the backup device in the correct order.

19.1.2 Volume Label Check

Use

You must first [initialize \[Page 91\]](#) tape volumes before you can use them to back up your Oracle database with BRBACKUP or BRARCHIVE. When BRBACKUP or BRARCHIVE writes to a volume, it first checks the volume label. The following graphic shows the volume label and check:

19.1 Volume Management



Activities

You can display the contents of a volume label as follows:

```
brbackup|brarchive -i show -n 1
```

The volume label is always the first file on a volume. This file has a format specific to BRBACKUP or BRARCHIVE and is written to the volume using `cpio`. The label file is called `.tape.hdr0`.

BRBACKUP or BRARCHIVE performs the following volume label checks:

1. Before BRBACKUP or 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 \[Page 98\]](#) 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 \[Page 459\]](#) 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 \[Page 480\]](#).

2. After a backup or archive to a volume is completed, the volume label is checked once more. This is to detect volume, tape 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 might be unusable if these errors are not recognized. The program checks whether the name of the database instance, the action ID, and the function ID match the current values.

19.1.3 Volume Expiration Period

Use

The volume expiration period for backups of the Oracle database is defined by the `init<DBSID>.sap` parameter [expir_period \[Page 459\]](#). This specifies the period in days during which a volume is locked, that is, cannot be used. When the period expires, you can reuse the volume for a backup. Volumes are normally tapes, though other backup media are possible.



`expir_period = 28` means that writing to a volume is possible 28 days after the volume was mounted and used. For example, if you use a volume on Monday 1st July, you cannot use it for another backup until Monday 29th July.

The start time of BRBACKUP or 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 (that is, 00:00 using the 24-hour clock) of the last day of the lock.



SAP recommends an expiration period of 28 days (the default value is 30 days).



If you set an 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 0.

Features

The **current** value of `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, where *n* is 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:

- Physical lock

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, that is, 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`.

- Logical lock

The internal information in the BRARCHIVE or BRBACKUP logs is decisive for a logical lock. The logs are updated when a database file has been backed up successfully. A

19.1 Volume Management

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 or BRARCHIVE logs and the value of `expir_period`. Under certain circumstances, discrepancies may occur between the physical and logical locks.

You can unlock volumes as follows:

- Unlocking a physically locked volume

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 – with [volume backup \[Page 481\]](#) or [volume archive \[Page 480\]](#) – but is rejected when the physical volume label check takes place. The volume can be reinitialized (with the same name) in order to cancel the physical lock, as follows:

- Temporarily set the `init<DBSID>.sap` parameter `expir_period` to 0, to circumvent the physical lock.
- Start BRBACKUP or BRARCHIVE, for example, as follows:

```
brbackup|brarchive -i -v <volume name>
```
- Reset the `expir_period` parameter to its previous value.

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

- Unlocking a logically locked volume

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 still want to use this volume before the logical lock has expired, you can switch off automatic volume management temporarily by mounting the volume on the backup device and entering the following command:

```
brbackup|brarchive -v SCRATCH
```

For more information, see:

- [brbackup -i \[Page 369\]](#)
- [brbackup -v \[Page 375\]](#)
- [brarchive -i \[Page 388\]](#)
- [brarchive -v \[Page 391\]](#)

19.1.4 Used Volumes

Definition

This section describes the tape volumes written by BRBACKUP or BRARCHIVE when you back up your Oracle database or archive the redo log files.

Structure

After successfully backing up the Oracle database with BRBACKUP and BRARCHIVE, the volumes produced contain the following files:

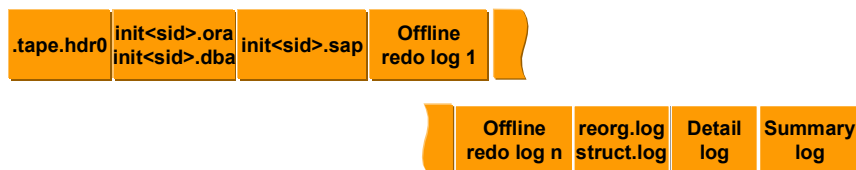
- Label: tape.hdr0 (position 1)
- init_oracle: Initialization files init<DBSID>.ora, init<DBSID>.dba (position 2)
- init_sap: Initialization file init<DBSID>.sap or the profile file defined under [-p/-profile \[Page 372\]](#) (position 3).
- Database files (data files, redo log files, control file)
- reorg_log: Main SAPDBA log reorg<DBSID>.log and structure log struct<DBSID>.log
- det_log: Detail log written by BRBACKUP or BRARCHIVE (second-to-last position)
- sum_log: Summary log written by BRBACKUP or BRARCHIVE (last position)

Tape Layout for BRBACKUP and BRARCHIVE

BRBACKUP:



BRARCHIVE:



Information about used volumes is contained in the BRBACKUP and BRARCHIVE logs:

- File system logs
 - Detail log, created for each backup
 - Summary log, extended during each backup (only for BRARCHIVE)
- Database logs
 - Table SDBAH with information about the whole backup
 - Table SDBAD with information about the backup of a file

Use

These log files should only contain information that was written by BRBACKUP or BRARCHIVE. Do **not** change this information manually. The information in database tables SDBAH and SDBAD is evaluated by the Computing Center Management System (CCMS), for example.

For more information, see [Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT \[Page 482\]](#).

19.1 Volume Management

For [automatic volume management \[Page 100\]](#), BRBACKUP uses the corresponding database log to select available volumes. BRARCHIVE uses the summary file system log. BRARCHIVE cannot rely on the database logs because it also runs when the database has been shut down.

19.1.5 Scratch Volume

Definition

When BRBACKUP or BRARCHIVE requests a scratch volume for backup, this means that you can use any volume for which the expiration period has ended. It does **not** mean that you have to mount a volume with the name `SCRATCH`.

Use

You can also initialize a volume with the name `SCRATCH` explicitly, for example, using `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. However, this situation might occur when a scratch tape is requested and a tape named `SCRATCH` is mounted.

19.1.6 Selecting Volumes Manually

Use

You can select the volumes for an Oracle database backup manually. You can do this if the BRBACKUP or BRARCHIVE [automatic volume management \[Page 100\]](#) is deactivated by using the reserved volume name `SCRATCH`. However, BRBACKUP or BRARCHIVE still checks the expiration period of the volume, and only allows you to use volumes for which the expiration period has ended.

For more information, see [Scratch Volume \[Page 98\]](#).

Prerequisites

- Make sure that initialized volumes are available for the backup.
- Determine whether the required expiration period has been configured in profile parameter [expir_period \[Page 459\]](#) and change the value when necessary.

Procedure

You can start the backup with BRBACKUP or BRARCHIVE in one of the following ways:

- Using the profile `init<DBSID>.sap`
 - a. Enter one of the following parameter values in profile `init<DBSID>.sap`:


```
volume_backup = SCRATCH
volume_archive = SCRATCH
```

For more information, see [volume_backup \[Page 481\]](#) and [volume_archive \[Page 480\]](#).
 - b. Start BRBACKUP or BRARCHIVE.
- Using the option `-v SCRATCH`

- a. Leave the profile unchanged (a volume pool can be defined).
- b. Start BRBACKUP or BRARCHIVE with the option `-v SCRATCH`.

Result

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 or BRARCHIVE volumes are accepted. See [Volume Expiration Period \[Page 95\]](#).



If you use scratch volumes, it might make sense to include the weekdays or days of the month in the volume names. This helps to make the names more meaningful.

19.1.7 Selecting Volumes with External Tools

Use

You can also use external tools to determine the names of the volumes relevant for the Oracle backup. This might involve an external volume management system or simply a shell script. You can do this if you have deactivated BRBACKUP or BRARCHIVE [automatic volume management \[Page 100\]](#) by calling them 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 make sure that only non-locked volumes are suggested for backup. Otherwise, BRBACKUP or BRARCHIVE 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 \[Page 459\]](#). Change the value if necessary.
2. Start BRBACKUP or BRARCHIVE with the option `-v <volume list>`. Before starting the backup, BRBACKUP or BRARCHIVE checks whether the mounted volumes agree with those in the volume list and whether the expiration period has expired.

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

You can select the volume names yourself by defining them in the call with the option `-v|-volume` (using the 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 or BRARCHIVE can use the following sample script to assign the volume names to the day of the month on which the backup was started.



The name `<DBSID><X><dd><n>` is made up of:
DBSID = ORACLE_SID (name of the database instance),

19.1 Volume Management

```

x = A for BRARCHIVE or x = B for BRBACKUP
dd = day of month
n = next volume number within one backup.
Here is the script:
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

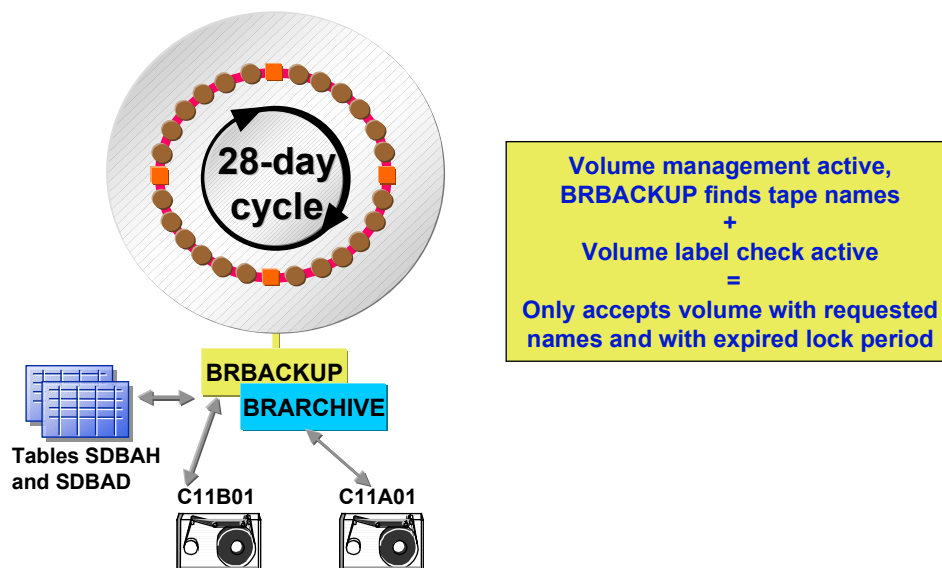
```

19.1.8 Selecting Volumes Automatically

Use

BRBACKUP or BRARCHIVE automatically selects 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.

Automatic Volume Management



Prerequisites

When you use automatic volume management, do not use weekdays or days of the month in the volume names, because BRBACKUP or BRARCHIVE does not check whether or not a holiday fell in the period since the last backup. 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 tape volumes.

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 \[Page 91\]](#) new volumes when necessary.
2. Determine whether the required [expiration period \[Page 95\]](#) has been configured in profile parameter [expir_period \[Page 459\]](#). Change the value when necessary.

19.2 Hardware Compression

3. Start the backup with BRBACKUP or BRARCHIVE. Only volumes that are **not** locked are selected from the pool of available volumes.
4. Mount the requested volumes. During the next run, BRBACKUP and BRARCHIVE attempts to write to all the available volumes in the pool in sequence.
5. You can display the names of the volumes required for the next volume as follows:

```
brbackup -q
```

```
brarchive -q
```

6. You can check whether you have mounted the correct volume in the backup device with:

```
brbackup -q check
```

```
brarchive -q check
```

These commands do **not** actually start the backup. In particular, you can use these options **before** scheduling a backup with CRON or a similar tool.

For more information, see [-q|-query \[Page 373\]](#).

If you want to [select the volumes manually \[Page 98\]](#) you can deactivate automatic tape management. To do this, set the `init<DBSID>.sap` parameters [volume_backup \[Page 481\]](#) and [volume_archive \[Page 480\]](#) to `SCRATCH` or call BRBACKUP or BRARCHIVE with the option `-v SCRATCH`. For more information, see [Scratch Volume \[Page 98\]](#).

19.2 Hardware Compression

Use

When backing up your Oracle database to tape, always use hardware compression if your tape devices support it. This reduces backup time because more data can be written to a single volume. Tape units with hardware compression are now industry-standard. The compression method used is normally based on the Lempel-Ziv algorithm. A few operating systems also support hardware compression for disk.

You can also use [software compression \[Page 102\]](#).

Features

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, but this can vary as follows:

- 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.

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, use [brbackup -k only \[Page 370\]](#) to approximate compression rates. Repeat this at least once a month to update the compression rates.

After a large data transfer or a reorganization of a tablespace, you must compress the affected tablespaces again. If a database file has no essential changes in two consecutive compression runs, you can consider the compression rate to be constant. You only need check the compression rate again after a longer period (for example, after a year). You can exclude these files from regular compression since the compression rate stays constant.

19.3 Software Compression



If you are using the BACKINT interface to an external backup tool, the above is not relevant and you do not need to do it.

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. You can use parallel compressions to determine the compression rates (that is, without starting a backup).

Activities

For hardware compression, you set the `init<DBSID>.sap` parameter `compress` to `hardware`. Be sure to 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).

Example

- Device type:
`backup_dev_type = tape|pipe|tape_auto|pipe_auto|tape_box|pipe_box`
- Addresses for the tape device:
`tape_address = (/dev/rmt/0hnc)`
`tape_address_rew = (/dev/rmt/0hc)`
- Compression parameters:
`compress = hardware`
- Tape size:
`tape_size = 16G`

See also:

[backup_dev_type \[Page 447\]](#)

[tape_address \[Page 475\]](#)

[tape_address_rew \[Page 477\]](#)

[compress \[Page 454\]](#)

[tape_size \[Page 479\]](#)

19.3 Software Compression

Use

When backing up your Oracle database to tape or disk, you can use software compression. Use software compression only if you have no tape devices with [hardware compression \[Page 101\]](#). Using hardware and software compression at the same time does **not** improve compression rates.

- Advantages
If you make remote backups over a network, using software compression significantly reduces the network load.
- Disadvantages
 - High CPU utilization

- Lengthy backups due to compression process

Features

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, assuming that the number of parallel copy processes was not reduced by changing the [exec_parallel \[Page 459\]](#) parameter. The tape devices must be defined in parameters [tape_address \[Page 475\]](#) and [tape_address_rew \[Page 477\]](#).

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.

For more information, see [Logs for BRBACKUP, BRARCHIVE, BRRESTORE and BRCONNECT \[Page 482\]](#).



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. For successful compression, make sure that the compression directory has at least as much free space as the largest database file needs before the compression.

If you specify `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).

Example

- Device type:
`backup_dev_type = tape|pipe|tape_auto|pipe_auto|tape_box|pipe_box`
- Tape device addresses:
`tape_address = (/dev/rmt/0mn)`
`tape_address_rew = (/dev/rmt/0m)`
- Compression parameters:
`compress = yes`
- Tape size:
`tape_size = 18G`

See also:

[backup_dev_type \[Page 447\]](#)

[tape_address \[Page 475\]](#)

[tape_address_rew \[Page 477\]](#)

19.3 Software Compression

[compress](#) [Page 454]

[tape_size](#) [Page 479]

20 Backup Methods

This section describes special methods to perform backup, archive, and recovery with the Oracle database:

- [Backup to Multiple Disks \[Page 105\]](#)
- [Backup to a Remote Disk \[Page 105\]](#)
- [Backup to a Remote Tape Device \[Page 108\]](#)
- [Two-Phase Backup \[Page 109\]](#)
- [Structure-Retaining Database Copy \[Page 111\]](#)
- [Parallel Backup \[Page 113\]](#)
- [Unattended Backup \[Page 114\]](#)
- [BRBACKUP and BRARCHIVE Backups in One Run \[Page 116\]](#)

20.1 Backup to Multiple Disks

Use

You can use [BRBACKUP \[Page 360\]](#) for Oracle backup to multiple disks if the space available on one disk or logical volume is not sufficient.

To do this, you use the `init<DBSID>.sap` profile parameter `backup_root_dir` to specify the directories on the different disks where you want to save your database files.



This is an example of the entries required in the [init<DBSID>.sap \[Page 445\]](#) initialization profile for parallel backup:

- 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 to back up the database files. The number of copy processes corresponds to the number of disks. Since BRBACKUP attempts to optimize the speed of the backup, all the hard disks specified in [backup_root_dir \[Page 450\]](#) are written to, except if the number of files you want to back up is smaller than the number of disks. You can change this setting with the `init<DBSID>.sap` parameter [exec_parallel \[Page 459\]](#) or the command option [-e|-execute \[Page 368\]](#).

See also:

[backup_dev_type \[Page 447\]](#)

[compress \[Page 454\]](#)

20.2 Backup to a Remote Disk

Use

20.2 Backup to a Remote Disk

You can perform an Oracle backup to a remote disk directly with [BRBACKUP \[Page 360\]](#) or [BRARCHIVE \[Page 383\]](#), or as part of an incremental backup using the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#).



We only recommend remote backup when your network is very fast and stable. Therefore, we do not recommend this procedure for production systems in most cases. However, you can use it to back up test systems.

Prerequisites

- The target directories for the backup defined in the parameter [stage_root_dir \[Page 471\]](#) and [archive_stage_dir \[Page 447\]](#) must exist on the remote host.
- The name of the remote host and the relevant user must be specified in the initialization profile with the parameters [remote_host \[Page 464\]](#) and [remote_user \[Page 464\]](#).
- If you use FTP as the transmission program, a password is needed for the remote host. Do one of the following for this:
 - Specify the password in the [remote_user \[Page 464\]](#) parameter.


```
remote_user = "<user_name> <password>"
```
 - Use the password of the BRBACKUP database user, which happens if you do not explicitly specify a password. Make sure that the password of the operating system user is the same.
- No password is needed if you use the RCP command for the remote disk backup. A prerequisite for a successful RCP call is, for example, the following entry in the `.rhosts` file, which is in the `HOME` directory of the remote user on the remote host:

```
<host_name> <user_name>
```

For more information, see your operating system documentation.

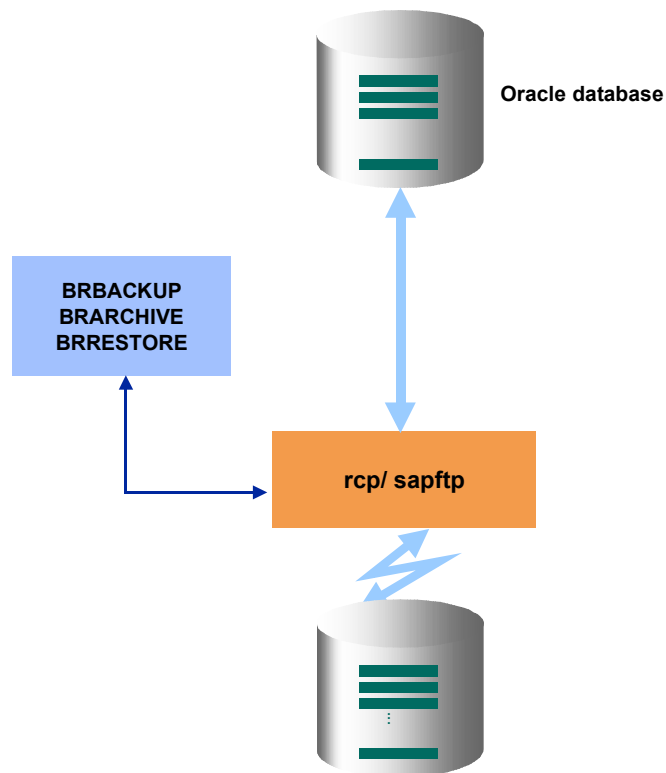
Features

Remote Disk Backup with BRBACKUP

A remote disk backup removes the need for a Network File System (NFS) disk mount, if the backup is across the network. The NFS protocol is relatively insecure, which means that the backup must be verified. In contrast, RCP and FTP are relatively secure ways of transferring data over the network, which means that you do not have to verify the backup.

You need to meet the prerequisites above and make the following entries in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

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



A remote backup to disk is particularly useful with the [standby database \[Page 162\]](#) and hierarchical storage management systems.

Incremental Remote Backup with RMAN

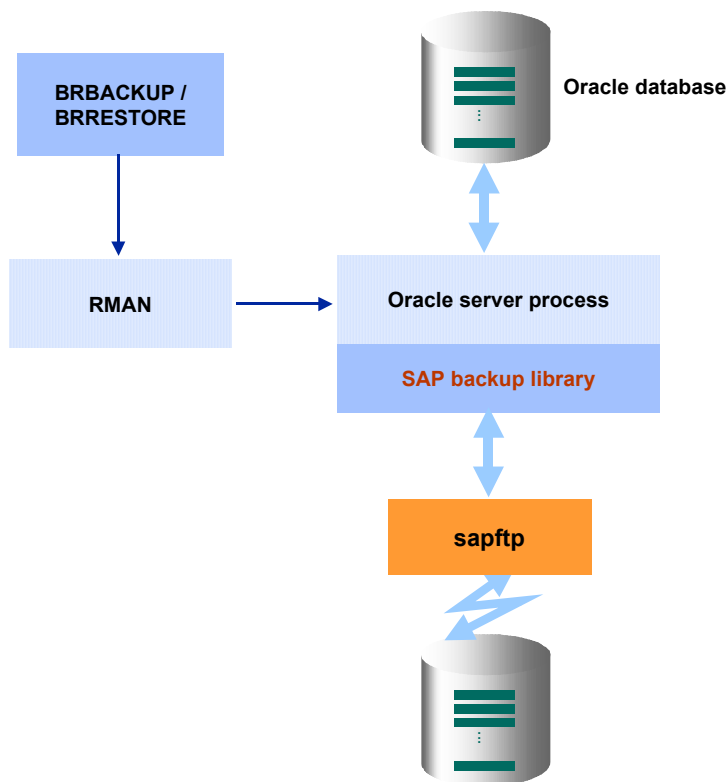
You can use [RMAN \[Page 137\]](#) to perform an incremental disk backup of a remote host.

You need to meet the prerequisites above and make the following entries in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

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

FTP is always used automatically as the transmission program for incremental backup to remote disk.

20.3 Backup to a Remote Tape Device



See also:

[RMAN Backup with the SAP Backup Library \[Page 142\]](#)

[Profile Parameters and BRBACKUP Command Options \[Page 380\]](#)

20.3 Backup to a Remote Tape Device

Use

With [BRBACKUP \[Page 360\]](#) and [BRARCHIVE \[Page 383\]](#) you can back up files of the Oracle database to a remote tape device that is connected to a UNIX host in the network. You can use several tape devices on the remote host for backup and these are used in parallel.

The UNIX versions of the local and remote hosts need not be identical. For example, if the database runs on an HP-UX host, you can perform the backup on an AIX host.



We only recommend remote backup when your network is very fast and stable. Therefore, we do not recommend this procedure for production systems in most cases. However, you can use it to back up test systems.

Prerequisites

- 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|rsh <host_name> date.`: only one line with the output of the date command should be displayed.

20.4 Two-Phase Backup

- For a successful remote shell call, check that there is an entry as follows in the `.rhosts` file, which is in the `HOME` directory of the remote operating system user on the remote host:

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

where:

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

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

For more information, see your operating system documentation.

Activities

The individual database files are transferred to the remote host using a remote shell call, called a “pipe.” You define the remote host with the `init<DBSID>.sap` parameter [remote_host \[Page 464\]](#) and the user with the [remote_user \[Page 464\]](#) parameter.

On the remote host the files are written to tape using the UNIX `dd` command. You define the `dd` command in the profile `init<DBSID>.sap`, as in the following example:



```
copy_in_cmd = "dd bs=64k if=$"
copy_out_cmd = "dd bs=64k of=$"
```

The number of parallel copy processes normally corresponds to the number of backup devices available. You can change this setting with the `init<DBSID>.sap` parameter [exec_parallel \[Page 459\]](#) or the command option [-e|-execute \[Page 368\]](#).

See also:

[Backup to a Remote Disk \[Page 105\]](#)

20.4 Two-Phase Backup

Use

As an alternative to a direct Oracle backup to tape you can perform a two-phase backup with full support from [BRBACKUP \[Page 360\]](#) or [BRARCHIVE \[Page 383\]](#). 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 restore from tape is required, the files are directly copied to the original directories by [BRRESTORE \[Page 395\]](#).



You can also perform the second phase of the two-phase backup with external tools – that is, backup programs, operating system tools, and so on. In this case, you have full responsibility for the complete and correct execution of this phase.



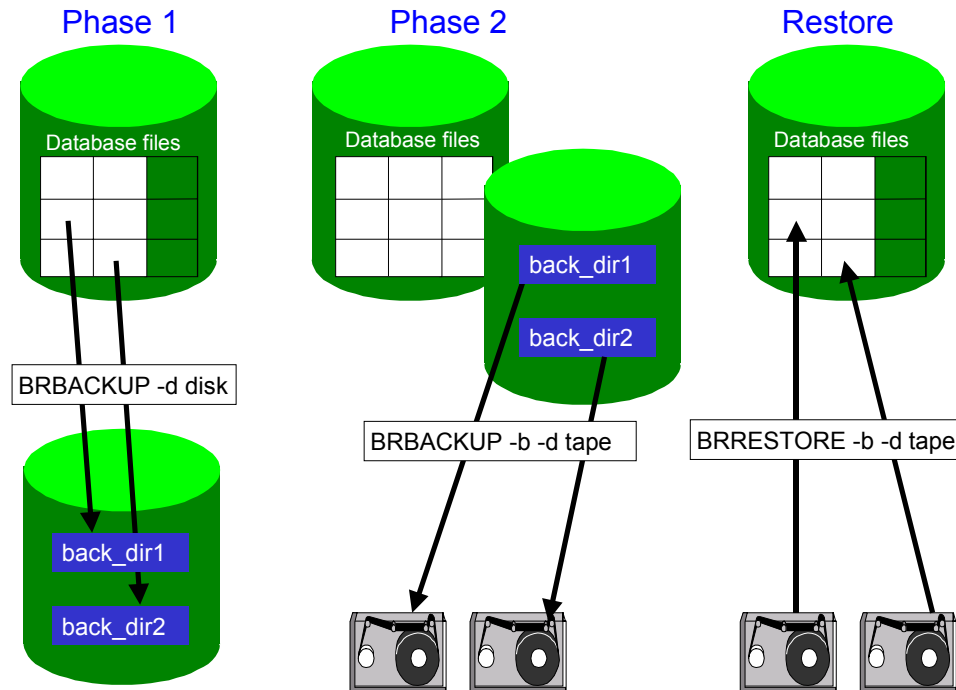
You can **not** use the BACKINT interface to copy backups from raw devices or compressed disk files to tape. If you use raw devices or compressed disk files, you must save directly to tape.

Features

Two-Phase Database Backup with BRBACKUP


20.4 Two-Phase Backup

- Phase 1
BRBACKUP backup to disk
- Phase 2
BRBACKUP backup from disk to tape
- Restore phase
BRESTORE restore from tape directly to the original directories or BRRESTORE from the disk to the original directories



To back up to tape (phase 2), start BRBACKUP with the relevant command option. For more information, see [-b|-backup \[Page 365\]](#).

Features of Two-Phase Database Backup with BRBACKUP

Advantages	Disadvantages
<ul style="list-style-type: none"> • The first phase can be much shorter than a direct backup to tape. • In the case of a recovery the restore phase is shortened, if the backup is directly available on 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 used in the second phase. 	<ul style="list-style-type: none"> • The hardware requirements (that is, disk storage and storage tapes) must be fulfilled. Additional disk storage space is required, compared to a direct tape backup. • You must start BRBACKUP twice for a two-phase backup. <div style="margin-top: 10px;">  <p>The backup type – that is, offline or online [Page 81] – and the extent of the backup [Page 77] – that is, complete or partial – must be identical in the first and second phases. In an offline backup the database remains</p> </div>

20.5 Structure-Retaining Database Copy

	open in the second phase.
--	---------------------------

Two-Phase Backup of Offline Redo Log Files with BRARCHIVE

Two-phase backup of the offline redo log files using BRARCHIVE runs analog to the BRBACKUP backup. However, you can only make a maximum of two copies to tape. We strongly recommend you to back up 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**

BRRESTORE restore from tape directly into the original directories or direct import of offline redo log files in an SAPDBA recovery from disk (no restore).

To perform the BRARCHIVE backup of the offline redo log files to a volume start BRARCHIVE with the appropriate command option. For more information, see [-al-archive \[Page 383\]](#).

Features of Two-Phase Backup of Offline Redo Log Files with BRARCHIVE

Advantages	Disadvantages
<ul style="list-style-type: none"> • The first phase can be much shorter than direct backup to tape, so that the SAPARCH directory is emptied more quickly. • In a recovery, the restore 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 restore. • 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 used in the second phase of the backup. 	<ul style="list-style-type: none"> • 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. • You must start BRARCHIVE twice for a two-phase backup.

20.5 Structure-Retaining Database Copy

Use

With [BRBACKUP \[Page 360\]](#) you can make a copy of the Oracle database files with exactly the same directory structure. You can use this type of database copy to:

- Generate a test system from a production system
- Set up a [Standby Database \[Page 162\]](#).
- Have a database backup available that saves you the restore during a recovery. In this case the Oracle Home directory is renamed as the new Oracle Home directory of the

20.5 Structure-Retaining Database Copy

database copy. The copied files are then the current files and you can apply the offline redo log files directly.

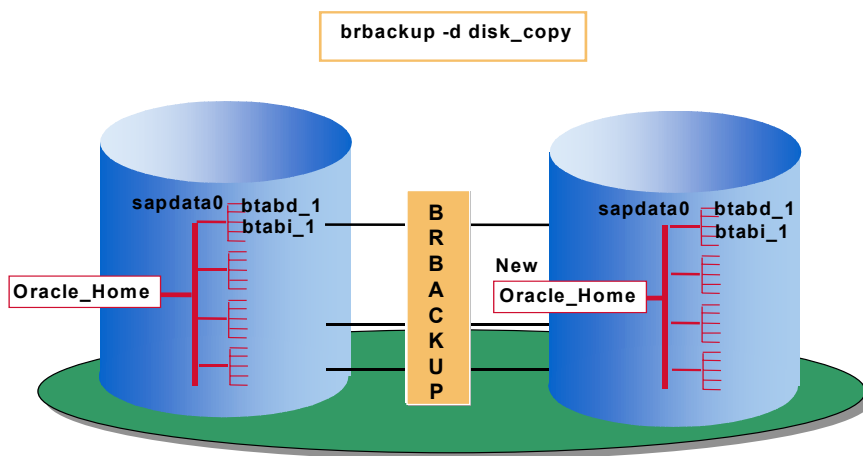
- Change the location of the database files. Database copy is the only way of moving database files between the file system and raw devices using BRBACKUP.

Prerequisites

You must create the following directories 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.



```
/oracle/C11/sapdata2/stabd_1/stabd.data1 is copied to  
/oracle/C12/sapdata2/stabd_1/stabd.data1
```



Since this is a one-to-one copy, software compression is **not** possible.

Activities

To copy the database, you have to define the name of the new database home directory (of the database copy) in the `init<DBSID>.sap` profile parameter [new_db_home \[Page 461\]](#) (for local disks) or [stage_db_home \[Page 471\]](#) (for remote disks). Also set the parameter [backup_dev_type \[Page 447\]](#) to

`disk_copy` | `disk_standby` | `stage_copy` | `stage_standby` or call up BRBACKUP with the relevant command option, for example, `brbackup -d|-device disk_copy`.



Under Windows, 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. For more information, see the BRBACKUP parameter [m|-mode \[Page 370\]](#).

20.6 Parallel Backup

Use

You can use [BRBACKUP \[Page 360\]](#) and [BRARCHIVE \[Page 383\]](#) to back up your Oracle database to several backup devices in parallel. The backup devices can be tape or disk drives. Parallel backup reduces the backup time and allows unattended operation (that is, backup in unattended mode).

Parallel backup is especially useful for [large databases \[Page 153\]](#). For more information, see [Parallel Backup of Large Databases to Disk with BRBACKUP \[Page 156\]](#).

Features

- Parallel backup is possible to local or remote backup devices.
- 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, using `compress = yes`. For more information, see [compress \[Page 454\]](#).
- When you perform parallel backups to several backup devices, BRBACKUP attempts to optimize the distribution of the database files among the backup volumes using load balancing, as follows:
 - BRBACKUP attempts to balance the load equally among all the backup devices, which is called time optimization. If this is not possible, it attempts to divide the dataset equally among the individual backup devices, which is called size optimization.
 - Whenever possible, BRBACKUP saves files from one disk 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). For more information, see [-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cds \[Page 390\]](#).
- The addresses of the tape devices are defined in the following `init<DBSID>.sap` parameters:
 - [tape_address \[Page 475\]](#)
 - [tape_address_rew \[Page 477\]](#)
 - [tape_address_arch \[Page 476\]](#)
 - [tape_address_rew_arch \[Page 477\]](#)
- 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 \[Page 450\]](#) or [stage_root_dir \[Page 471\]](#).



This is an example of the entries required in the [init<DBSID>.sap \[Page 445\]](#) initialization profile for parallel backup:

- Device type:
`backup_dev_type = tape|disk|pipe|stage`
- Addresses for tape devices:
`tape_address = (/dev/rmt/0hnc, /dev/rmt/1hnc)`
`tape_address_rew = (/dev/rmt/0hc, /dev/rmt/1hc)`

20.7 Unattended Backup

- Addresses for directories on hard disk:
`backup_root_dir = (/backup/dir1, /backup/dir2)`
- Compression parameters:
`compress = no|software|hardware`
- Tape size:
`tape_size = 18G|16G`

- The number of parallel copy processes normally corresponds to the number of backup devices available. You can change this setting with the `init<DBSID>.sap` parameter [exec_parallel \[Page 459\]](#) or the command option `-e|-execute [Page 368]`.

See also:

[backup_dev_type \[Page 447\]](#)

[tape_size \[Page 479\]](#)

20.7 Unattended Backup

Use

With [BRBACKUP \[Page 360\]](#) and [BRARCHIVE \[Page 383\]](#), you can back up your Oracle database without monitoring or operator intervention. How you do this depends on your operating system. For more information on backup with Microsoft Windows, see [BRBACKUP/BRARCHIVE \[Page 512\]](#).

Features

Unattended Parallel Backup

You can make unattended backups if you have enough backup devices. This means that you need as many backup devices as volumes are required for the backup. BRBACKUP can then back up to these devices in parallel, without operator intervention to change the volumes. For more information, see [Parallel Backup \[Page 113\]](#).

Automatic Tape Changers

If you want to use automatic tape changers, you must define the [rewind_offline \[Page 466\]](#) parameter appropriately and set [backup_dev_type \[Page 447\]](#) to `tape_auto` or `pipe_auto`. For more information, see [Backup with Automatic Tape Changers \[Page 176\]](#).

Serial Backup

Unattended backup is also possible if fewer parallel copy processes than connected backup devices are possible (for example, due to the impact on performance). To do this, set the parameter [exec_parallel \[Page 459\]](#) to 1. 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.

To make a parallel or serial backup on several tape devices, you must define the addresses of the backup devices in the following `init<DBSID>.sap` parameters:

- [tape_address \[Page 475\]](#)
- [tape_address_rew \[Page 477\]](#)
- [backup_root_dir \[Page 450\]](#) for a local disk backup
- [stage_root_dir \[Page 471\]](#) for a remote disk backup



Here is an example of these parameters:

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

Backups with BRBACKUP or BRARCHIVE 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, using command option [-a|-archive \[Page 364\]](#).

For more information, see [BRBACKUP and BRARCHIVE Backups in One Run \[Page 116\]](#).

Backup with CRON

For a successful unattended backup using CRON, you must meet the following requirements:

- The parameters in `init<DBSID>.sap` must be set correctly.
- The Crontab entries must be defined under user `root`.
- BRBACKUP or 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.
- Enough volumes (for example, tape or disk volumes) must be available, as well as a sufficient number of backup devices (if you want to perform a parallel backup).
- The correct tapes must be mounted in the backup devices. The operator cannot change the tapes during the backup run.

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.

For more information, see [brbackup -q \[Page 373\]](#) and [brarchive -q \[Page 389\]](#).

If automatic tape management is not active, mount tapes for which the expiration period has passed.

Examples



Online backup of the complete database
Unattended operation
Daily, Monday through Friday
Backup to start at 22:00

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

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

```
su - ora<dbsid> -c "brbackup -t online -c force -u"%system/<password>
```

or under `<sapsid>adm`:

```
su - <sapsid>adm -c "brbackup -t online -c force -u /"
```



Offline backup of the complete database

20.8 BRBACKUP and BRARCHIVE Backups in One Run

Unattended operation
Daily, Monday through Friday
Backup to start at 22:00
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 might look as follows:

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



Offline backup of the complete database
Unattended operation
Daily, Monday through Friday
Backup to start at 22:00
No 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 might look as follows:

```
su - ora<dbssid> -c "brbackup -t offline_force -c force -u" <<END
system/<password>
END
```



Archiving the offline redo log files
Unattended operation
Daily, Monday through Friday
Backup to start at 08:00
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<dbssid> -c "brarchive -ssd -c force -u"%system/<password>
```

20.8 BRBACKUP and BRARCHIVE Backups in One Run

Use

You can execute the database backup using [BRBACKUP \[Page 360\]](#) and the backup of the offline redo log files using [BRARCHIVE \[Page 383\]](#) in a single run. You can use this option to make more effective use of the increasing capacity of storage devices, such as tapes and

20.8 BRBACKUP and BRARCHIVE Backups in One Run

disks. It also makes an unattended backup easier, since after a backup with BRBACKUP, the BRARCHIVE run is started automatically. You can now perform this procedure in the DBA Planning Calendar of the Computing Center Management System (CCMS)

There are the following options for performing the backup in one run:

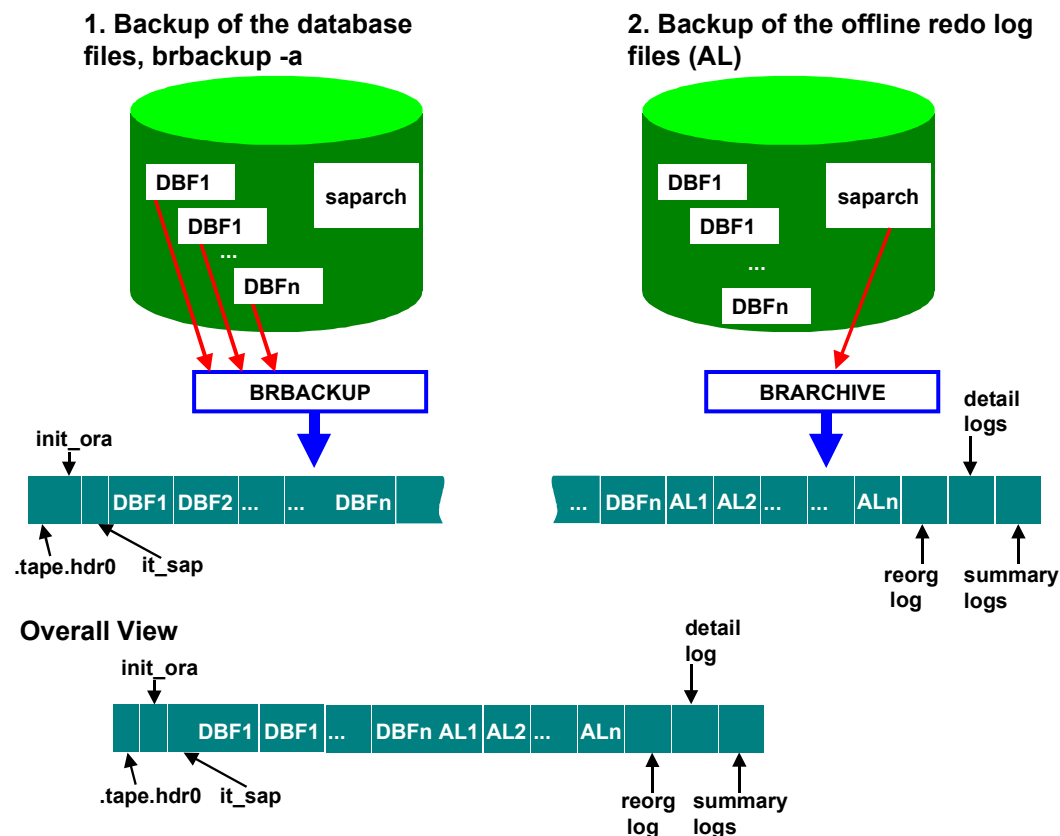
- BRBACKUP starts BRARCHIVE, using `brbackup -a`. For tape backups, the tapes are managed by BRBACKUP. For more information, see [-a|-archive \[Page 364\]](#).
- BRARCHIVE starts BRBACKUP, using `brarchive -b`. For tape backups, the tapes are managed by BRARCHIVE. For more information, see [-b|-backup \[Page 386\]](#).

We recommend the first option (`brbackup -a`). For tape backups, BRBACKUP uses the tapes defined in [volume backup \[Page 481\]](#). To execute the backup, BRBACKUP:

- Checks the volume label
- Backs up the tape header files (`.tape.hdr0`, `init_ora`, `init_sap`)
- Backs up the database files (but does not save logs)
- Calls BRARCHIVE

Then BRARCHIVE:

- Writes the offline redo log files to tape after the backed-up database files (without checking the label and without header files)
- Backs up all logs (that is, for BRBACKUP and BRARCHIVE)



20.9 Backup Verify



This is an example of a BRBACKUP command for 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
```

For more information, see BRBACKUP command [-a|-archive \[Page 364\]](#).

If you run the tape administration using [brarchive -b \[Page 386\]](#), the tapes defined in [volume archive \[Page 480\]](#) 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.



This is an example of a BRARCHIVE command for 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
```

For more information, see BRARCHIVE command [-b|-backup \[Page 386\]](#).

You can also perform a combined BRBACKUP and BRARCHIVE run to several tape devices in parallel. For more information, see [Parallel Backup \[Page 113\]](#).



Starting BRBACKUP and BRARCHIVE in sequence with BACKINT, or to disk, only means that BRBACKUP and BRARCHIVE run together. 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.

20.9 Backup Verify

Use

When making database backups, you must regularly perform a verify.



If you fail to verify backups, you might find that a backup cannot be used in the event of a database restore. Even if a backup is reported as successfully completed, this does **not** mean that it is always error-free.

There are the following types of verify:

- Backup tape readability

This verify checks the backup media (that is, usually tape). It checks the size of the backup files and whether the data can be read. In some cases, it also compares the backed-up data with the contents of the database byte-by-byte.

- Database block consistency

This verify checks the database itself block-by-block.

At least once in every [backup cycle \[Page 84\]](#), you must perform **both** types of verify. If possible, perform verify once a week, or even for every backup.

Integration

- You can perform verify at the command level of the [BRBACKUP \[Page 360\]](#), [BRARCHIVE \[Page 383\]](#), and [BRRESTORE \[Page 395\]](#) tools, as described below.

- You can also perform verify using the [action patterns \[Ext.\]](#) of the [DBA Planning Calendar \[Ext.\]](#).

Prerequisites

- Verify adds considerably to backup run times.
- Verify is always performed on complete backup volumes. A volume is first written and then verified.

Features

Verify of Backups with BRBACKUP

This type of verify only checks the backup tape readability. The method and extent of a [BRBACKUP \[Page 360\]](#) verify differs according to whether the backup is [online or offline \[Page 81\]](#):

- Offline backup
After the backup the files are copied back to a temporary directory, `compress_dir`, and the contents are compared to the original data in the database byte by byte.
- Online backup
After the backup the files are copied back to a temporary directory, `compress_dir`, but only the readability file sizes are compared. The byte-by-byte compare used in an offline backup is not possible with an online backup because the database is constantly changing as updates continue.

For more information, see [brbackup -w|-verify \[Page 376\]](#).

Independent Verify of Backups with BRRESTORE

This type of verify with [BRRESTORE \[Page 395\]](#) only checks the backup tape readability, separately from the backup, at a later time if you want. You can also perform this type of verify on another computer.

BRRESTORE only checks whether the backup can be read and its size, not the contents. The files are only read, not restored.

For more information, see [brrestore -w|-verify \[Page 405\]](#)

Verify of Backups with BRARCHIVE

This type of verify with [BRARCHIVE \[Page 383\]](#) only checks the backup tape readability. 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 byte by byte.
- `brarchive -sd|-scd|-ssd|-cds`
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.

For more information, see [-brarchive -w|-verify \[Page 391\]](#)

Verify of Backups with Oracle DBVERIFY

The Oracle DBVERIFY tool is available for both types of verify, that is, backup tape readability (but without a byte-by-byte comparison) and database block consistency. This

20.9 Backup Verify

means you can recognize errors early (for example, ORA-1578), before they lead to the termination of a program in an application that accesses the blocks.

You can use DBVERIFY as follows:

- Database backup with subsequent restore to a temporary directory (`compress_dir`) and check on the Oracle block structure:

```
brbackup -w use_dbv
```

- Online check of block structure on the database files without backup:

```
brbackup -w only_dbv
```

Any number of these verify processes can run in parallel, using parameter `exec_parallel`, option `-e`.

- Temporary restore of a database backup (`compress_dir`) and verify of the database block consistency:

```
brrestore -w use_dbv
```

The restore implicitly checks the readability of the backup.

Verify of BACKINT Backups

You can verify third-party backups using the BACKINT interface. This function confirms that backups are known to the external backup tool (that is, confirmation) and they are accepted when restoring. In this case there is no physical data check on the backup medium. Only an internal catalog of BACKINT backups of the backup tool is checked (that is, logical verification). Verifications using Oracle DBVERIFY are no longer supported in this context.



If you want to include verification of the internal Oracle block consistency with DBVERIFY, you can check the database directly without starting the backup. To do this, call BRBACKUP as follows:

```
brbackup -w only_dbv
```

From Release 4.5A, the external backup tool – that is, `backup_dev_type = rman_util` – can also be accessed using the Oracle Recovery Manager (RMAN). This gives you a complete range of functions for verification:

- The RMAN validate function is used for verifying backups. In this case, data is physically read by the backup medium and checked by RMAN for consistency.
- The verification with DBVERIFY is no longer required because for each backup all saved database files are checked by RMAN for internal Oracle block consistency.

You can also use BR programs from Release 4.5 for RMAN backups for lower SAP releases. Oracle 8.0 is required in this case. For more information, see Note 12741.



Of course, you can perform BACKINT backup verification directly using the backup tool (if supported). Some backup tools offer special parameters in the BACKINT parameter file, `init<SID>.utl`, that you can use to activate the verification independent of the BR option `-w|-verify`. For additional information, consult your BACKINT provider.

21 Approach to Restore and Recovery

Purpose

[SAPDBA \[Page 184\]](#) supports the recovery of the database after:

- Media errors, such as a disk crash or accidental deletion of a database file
- User errors such as software problems or when a user accidentally drops a table

This section tells you how to work out the best approach to get your database up and running again.



Always analyze the problem carefully before attempting to recover your database. If in doubt, seek support from SAP. The business costs of an incorrect or delayed recovery can easily outweigh consultancy fees.



We strongly recommend you to practice restore and recovery on a test system as similar as possible to your production system. Repeat this regularly, especially after you have modified the production system.

The SAPDBA restore and recovery functions are based on the SAP tools [BRBACKUP \[Page 360\]](#), [BRARCHIVE \[Page 383\]](#), and [BRRESTORE \[Page 395\]](#):

- During a restore, BRRESTORE copies database files from the backup medium to disk
- During a recovery, SAPDBA applies the data in the backups of the offline redo log files to the restored database files

Prerequisites

Try to locate:

- The backups of the missing or faulty data files made with BRBACKUP or BRARCHIVE.
- The [BRBACKUP logs \[Page 377\]](#) and [BRARCHIVE logs \[Page 392\]](#), which are very important for the restore because they contain all the information about the backups, such as directories, volumes, and timestamps. SAPDBA uses the logs to find the backups of the data files and the offline redo log files.



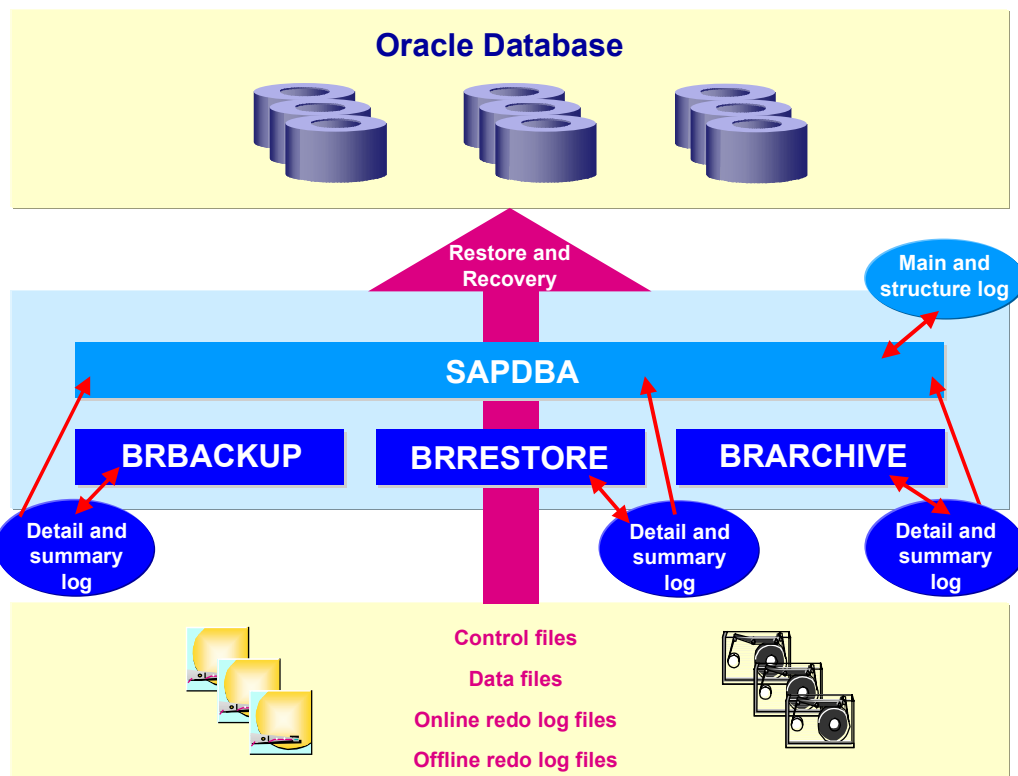
If you lose the entire database system, including such items as the BRBACKUP and BRARCHIVE logs and profiles, you need to perform [Disaster Recovery \[Page 123\]](#).



SAPDBA recovery is not based on BRBACKUP or BRARCHIVE logs in database tables, because these tables are not available when you recover the database.

You might also need to meet other prerequisites, depending on the context of the recovery.

The following graphic shows an overview of the restore and recovery process:



Process Flow

1. To analyze the problem, check the:
 - Database alert log
 - Trace files belonging to the background processes in the directory
\$ORACLE_HOME/saptrace/background.
 For more information, see [Error Analysis \[Page 488\]](#).
2. Ask yourself the following questions:
 - What is the status of the database? Is it available or not?
 - What kind of error has occurred? A media or a user error?
 - Which files are corrupt?
 - What type of file is affected? Data files, control files, online redo log files?
 - If a media error has occurred, is software or hardware mirroring available?
 - Do you have a [standby database \[Page 162\]](#)?
3. If a user error has occurred, the database is still available, and you have enough time, perform a [complete \[Page 83\]](#), [offline backup \[Page 81\]](#) before starting to restore and recover the database.
4. If a media error has occurred, you must replace the affected equipment and recreate the file system as it existed before the error.
5. Decide what kind of restore and recovery you want to perform:

Scenario	Aim	Solution
Loss of entire database	You want to recover as much as	Disaster Recovery [Page

system	possible	123]
Media error such as a disk crash	You want to recover to the time of failure	Partial restore and complete recovery with SAPDBA [Page 323]
User or software error	You want to recover to a selected point in time	Full restore and point-in-time recovery [Page 335]
Either of the following applies: <ul style="list-style-type: none"> Error in which all copies of redo log files are lost, but the database file backup is available You performed a full offline backup immediately before the error, such as during a software upgrade. 	To reset the database to the state it was in at the most recent offline backup. No recovery is possible, because the redo log files are not available.	Database reset [Page 340]



Only if you are very experienced, you might want to consider the following:

- [Restore of one tablespace \[Page 341\]](#)
- [Restore of individual files \[Page 341\]](#)

21.1 Disaster Recovery

Purpose

If you lose your entire Oracle database system (possibly including hardware), and have not taken any special security precautions – such as setting up an [Oracle standby database \[Page 162\]](#) – then you have to recover the system as much as possible, step by step. This section describes how to re-install the system, and how to keep data loss to a minimum by using [BRRESTORE \[Page 395\]](#).

Prerequisites

The following information is helpful to understand this section:

	UNIX	Windows
Environment variables	\$ORACLE_HOME \$SAPDATA_HOME	%ORACLE_HOME% %SAPDATA_HOME%
Oracle Server Manager	Oracle 8.1: svrmgrl	Oracle 8.1: SVRMGRL
Oracle SQLPLUS	Oracle 9.0: sqlplus	Oracle 9.0: SQLPLUS
Profile file directory	\$ORACLE_HOME/dbs	%ORACLE_HOME%\database

The default values for these variables are as follows:

	UNIX	Windows
Oracle 8.1 and 9.0	\$ORACLE_HOME = /oracle/<SID>/<DB_VERS>	%ORACLE_HOME% = <x>:\orant
	\$SAPDATA_HOME =	%SAPDATA_HOME% =

21.1 Disaster Recovery

	/oracle/<SID>	<y>:\oracle\<SID>
--	---------------	-------------------



Windows: If the SAPARCH, SAPBACKUP, or SAPREORG directories are not located under %SAPDATA_HOME%, use the corresponding environment variables %SAPARCH%, %SAPBACKUP% or %SAPREORG%.

Process Flow

1. You recover the hardware environment and operating system.

Install the same operating system that you had before the loss.

Recreate 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 documentation, which you can find at <http://service.sap.com/instguides>. Keep the reserved memory for superuser (root) to a minimum, as recommended in the installation guide.

2. You install the database and SAP System software.

The installation of the SAP software includes the software installation of the database. The initial SAP database should be completely new. However, do not load any data into the empty SAP tables. Alternatively, restore a full backup of your operating system or use the SAP installation CD.

3. You create the file systems.

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. You adapt the [initialization profile init<DBSID>.sap \[Page 445\]](#).

At least one valid `init<DBSID>.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).

Also note the following points:

- Check all parameters specific to the operating system and adapt them. Configure the same blocksize in the [cpio in flags \[Page 457\]](#) parameter and – if you used the `dd` commands for backup – the [dd in flags \[Page 457\]](#) parameter.
- To retrieve the latest backup of the BRBACKUP profile file `$ORACLE_HOME/dbs/init<DBSID>.sap`, enter the following commands (after adapting the parameters specific to the operating system):

```
cd $ORACLE_HOME/dbs
brrestore -n init_sap
```

Alternatively the `init<DBSID>.sap` file can also be read as a third file from a BRBACKUP tape using the `cpio` operating system command.



UNIX:

```
cd $ORACLE_HOME/dbs
mt -f /dev/... rewind
mt -f fsf 2
cpio -ivub < /dev/...
```

Windows:

```
cd %ORACLE_HOME%\database
mt -f /dev/... rewind
mt -f /dev/... fsf 2
cpio -ivub -I /dev/...
```



If you have configured a different block size in the `init<DBSID>.sap` parameter `cpio_flags`, you must use this same block size when restoring.

5. You restore other profile and log files.

a. Restore the files from the last BRBACKUP database backup, listed below.

- Oracle profile file: `$ORACLE_HOME/dbs/init<DBSID>.ora`
SAPDBA profile file: `$ORACLE_HOME/dbs/init<DBSID>.dba`

Do this as follows:

```
cd $ORACLE_HOME/dbs
brrestore -n init_ora
```

(this includes both profiles)

- SAPDBA central log: `$SAPDATA_HOME/sapreorg/reorg<DBSID>.log`
SAPDBA structure log: `$SAPDATA_HOME/sapreorg/struct<DBSID>.log`

Do this as follows:

```
cd $SAPDATA_HOME/sapreorg
brrestore -n reorg_log
```

(this includes both logs)

- BRBACKUP summary log: `$SAPDATA_HOME/sapbackup/back<DBSID>.log`

Do this as follows:

```
cd $SAPDATA_HOME/sapbackup
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.

- BRBACKUP detail log `$SAPDATA_HOME/sapbackup/<coded timestamp>.<fid>`

Do this as follows:

```
brrestore -n det_log
```



From SAP Release 4.5B, you can quickly restore all logs from the BRBACKUP tape into the current directory using a single BRRESTORE command:

21.1 Disaster Recovery

```
brrestore -n all_log
```

Then copy the individual logs into the correct directories.

- b. 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.
- c. 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.
- d. Restore the following file from the BRARCHIVE tape:

BRARCHIVE summary log: `$SAPDATA_HOME/saparch/arch<DBSID>.log`

Do this as follows:

```
cd $SAPDATA_HOME/saparch
```

```
brrestore -n sum_log
```

- e. Remove the BRARCHIVE tape from the tape unit and replace it with the BRBACKUP tape.

If you use the BACKINT interface the procedure differs in some respects:

The profiles `init<DBSID>.utl` (optional, depending on backup program) and `init<DBSID>.sap` must exist before you use BACKINT. In the profile `init<DBSID>.sap` you only have to define the name of the profile of the external backup program in the parameter `util_par_file`. You might have to reconstruct this first. You can do this either by editing, or by using the BACKINT command line interface to restore it directly.

- f. Then restore the following files individually:

- BRBACKUP summary log:

`$SAPDATA_HOME/sapbackup/back<DBSID>.log`

Do this as follows:

```
cd $SAPDATA_HOME/sapbackup
```

```
brrestore -d util_file -b2 "#NULL" -m  
$SAPDATA_HOME/sapbackup/back<DBSID>.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 detail log:

`$SAPDATA_HOME/sapbackup/<coded_timestamp>.<fid>`

Do this as follows:

```
brrestore -d util_file -b2 "#NULL" -m  
$SAPDATA_HOME/sapbackup/<coded_timestamp>.<fid>
```

- You can restore all other profiles and log files in the same way.

Do this as follows:

```
brrestore -d util_file -b2 "#NULL" -m <pathname>
```

where `<pathname>` is to be replaced by:

21.2 Partial Restore and Complete Recovery

```

$ORACLE_HOME/dbs/init<DBSID>.ora
$ORACLE_HOME/dbs/init<DBSID>.sap
$ORACLE_HOME/dbs/init<DBSID>.dba
$ORACLE_HOME/dbs/init<DBSID>.utl
$$SAPDATA_HOME/saparch/arch<DBSID>.log
$$SAPDATA_HOME/sapreorg/struct<DBSID>.log
$$SAPDATA_HOME/sapreorg/reorg<DBSID>.log

```



The procedure described above is complex. You can use the restore function of the external tool as an alternative.

6. You reset the database to the last online or offline backup.



An online backup must always be recovered to a point in time after the backup.
If you require a more current status than that of the restored offline backup you can also perform a recovery.

Restore the backup completely:

```
brrestore -b <coded_timestamp>.<fid> -m full
```

The <coded_timestamp> and <fid> were already determined from the summary BRBACKUP log.



If you have to perform recovery you must **not open** the database at this point.

7. For a **recovery** you perform the following steps:
- Remove the BRBACKUP tape from the tape unit and replace it with the BRARCHIVE tape.
 - Restore the required offline redo log files:

```
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<DBSID>.log).

- Perform database recovery with the Oracle SQLPLUS:

```

SQLPLUS> connect / as sysdba
SQLPLUS> startup database
SQLPLUS> recover database using backup controlfile until
cancel
SQLPLUS> alter database open resetlogs;

```

21.2 Partial Restore and Complete Recovery

Use

This section tells you about restoring damaged or lost data files after a failure in your Oracle database, and then recovering the database to the time of failure. You normally do this after a media error, such as a disk crash. With this function you can:

21.2 Partial Restore and Complete Recovery

1. Restore lost data files by using appropriate backups
2. Recover the restored data file to the time of failure using the redo log files

This function consists of a number of phases that are executed either manually or automatically by SAPDBA, in a predetermined sequence.

Integration

This function performs a **complete** recovery to the time of failure. If you want to perform a **point-in-time** recovery – that is, a recovery to some time other than the time of failure – see [Full Restore and Point-In-Time Recovery \[Page 130\]](#).

Prerequisites

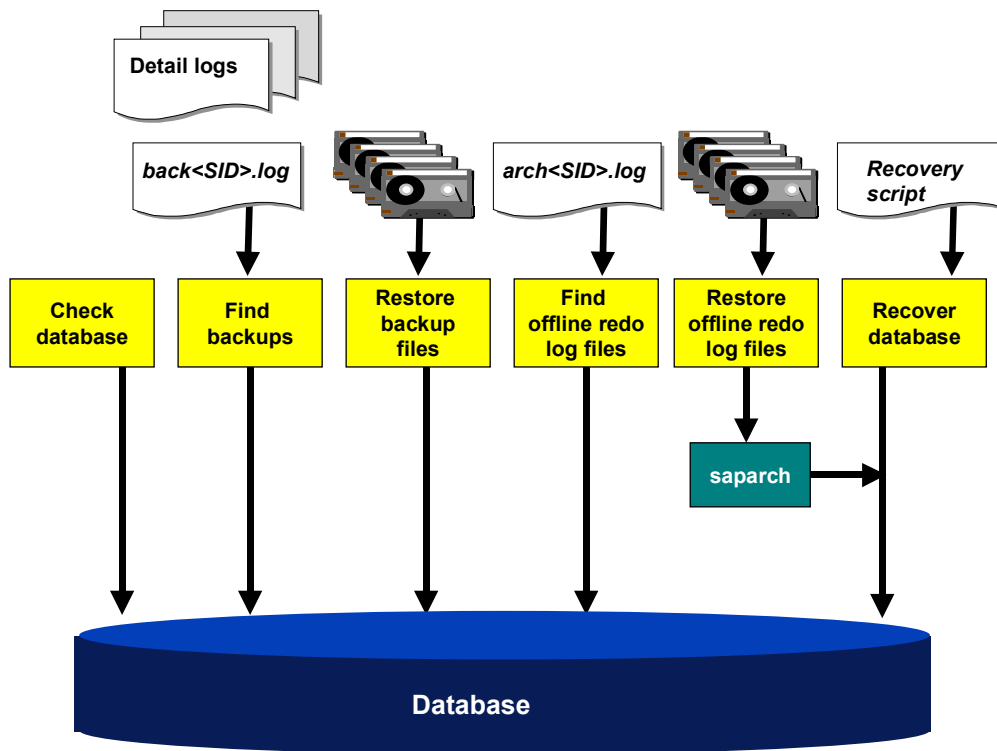
To finish a partial restore and complete recovery, you must have the files shown in the following table, either as originals or backups. The table shows what you must do if the files are **not** available:

Missing File Type	Solution if Unavailable
Database files	Use the most recent database backup available. You must always start by restoring a backup.
Offline redo log files	Recover to the most recent available redo log file. If none available, perform a database reset from an offline backup or an online consistent backup.
init<SID>.dba and init<SID>.ora	Restore from tape.
Control files	Copy one of the control file mirrors
BRBACKUP logs [Page 377] and BRARCHIVE logs [Page 392]	Restore from last backup.

For more information if you have lost particular files, see [Restoring Individual Files \[Page 341\]](#).

Features

The following graphic shows how a partial restore and complete recovery with SAPDBA works:



Activities

1. In the Check Database phase, SAPDBA checks the status of all files in the database (that is, the control files, online redo log files, and data files) as well as the tablespace status (online/offline; online backup mode). SAPDBA does the following:
 - a. SAPDBA refers to entries in Oracle's V\$Views, such as V\$DATAFILE, V\$RECOVER_FILE. If an error is detected during this phase, a safe check must be performed. That is, the database must be shut down, using `shutdown immediate`. If this is unsuccessful, SAPDBA suggests `shutdown abort`.
 - b. To update the V\$Views, SAPDBA sets the database to *mount* status. SAPDBA logs any recorded errors in data files in the `sapreorg` directory with the `rcv` suffix (for recovery.) A safe check is a prerequisite for any subsequent restore and recovery activities.
 - c. SAPDBA does not create missing sapdata directories automatically. However, SAPDBA automatically creates missing subdirectories.
2. In the Find Backup Files phase, SAPDBA determines the required backups using the entries in the BRBACKUP summary log file `back<SID>.log` (return code 0 or 1). The associated detail logs show whether the required data files were in the backup. The data files can be compiled from various backups. To minimize the subsequent recovery time, SAPDBA always suggests the most recent backup.
3. In the Restore Backup Files phase, SAPDBA calls BRRESTOE to restore the data files to their original location. If only index files are missing, SAPDBA can recreate and build up these files using Database Dictionary information.
4. In the Find Offline Redo Log Files phase, SAPDBA determines the offline redo log files required for a complete recovery. The BRARCHIVE summary log file `arch<SID>.log` lists the tapes where the offline redo log files have been saved. You can choose between a

21.3 Full Restore and Point-In-Time Recovery

first or second backup (for example, when saved, with `brarchive -cds`). SAPDBA takes existing online redo log files and offline redo log files in `saparch` into consideration. The Find Archive phase ends when the appropriate backups have been found for all required offline redo log files.

5. In the Restore Offline Redo Log Files phase, SAPDBA calls `BRRESTORE` to read the offline redo log files that have been found from tape back to the `saparch` directory.
6. In the Recover Database phase, SAPDBA creates recovery scripts in a subdirectory of `sapreorg`. Using these scripts, SAPDBA saves a control file, and sends a recover database statement (that is, a complete recovery) to the database. The SAPDBA message *Recover database terminated successfully* indicates that the database has been recovered completely.

21.3 Full Restore and Point-In-Time Recovery

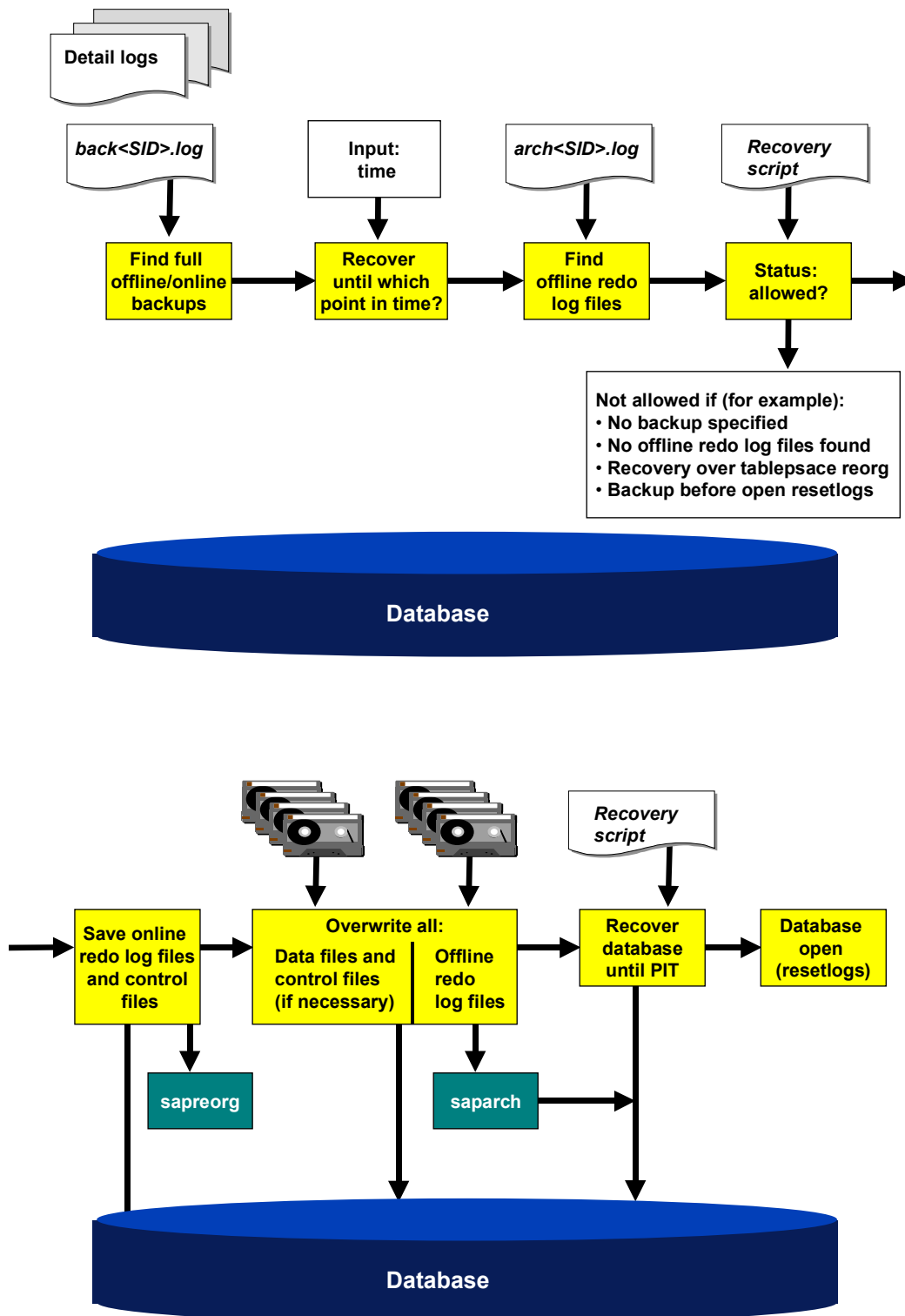
Use

You can use this function to fully restore and then recover your Oracle database to a specified point in time (PIT). You normally use this function when there has been a logical error – that is, a user or software error – and you want to recover the database to the point immediately before the error. In this way, you minimize lost data.

Prerequisites

- We recommend you to:
 - Perform a [full offline or online backup \[Page 81\]](#). If the database is running, use SAP tools, otherwise use operating system tools.
 - Back up all offline redo log files using `BRARCHIVE`. For more information, see [-a/-archive \[Page 385\]](#).
- It must be possible to set the database to status mount or open.
- You must have the following data available:
 - The [BRBACKUP logs \[Page 377\]](#) and the [BRARCHIVE logs \[Page 392\]](#)
 - The data file backups
 - All offline redo log files between the data backup and the chosen PIT

Features



Activities

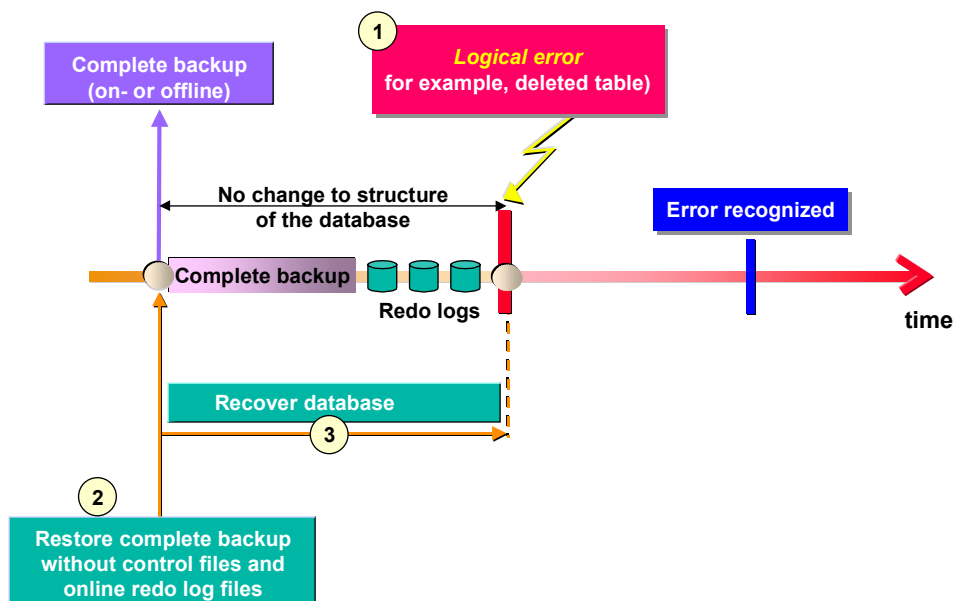
21.3 Full Restore and Point-In-Time Recovery

1. You select a full online or offline backup. SAPDBA uses the BRBACKUP summary log file `back<DBSID>.log` and the corresponding detail logs to find the correct backups.
2. You enter the recovery end-point.
3. You select the required offline redo log files. SAPDBA uses the entries in the BRARCHIVE summary log file, `arch<SID>.log` to find the correct files.
4. SAPDBA indicates whether your intended recovery is permitted. It rejects the recovery if:
 - No **full** backup has been specified or it could not find the required offline redo log files.
 - The period of your intended recovery contains a [tablespace reorganization with data files \[Page 271\]](#).
 - The selected backup is dated before the last time the database was opened using the `resetlogs` option.
5. SAPDBA saves the current online redo log files and a control file to a subdirectory of `sapreorg`.
6. SAPDBA calls BRRESTORE to restore all data files from the backup medium. If a tablespace was extended at the time of the recovery, it also restores the control files.
7. SAPDBA calls BRRESTORE to restore the offline redo log files.
8. Using the recovery script, SAPDBA recovers the database to the required PIT.
9. SAPDBA opens the database using the `resetlogs` option and prompts you to perform a backup.
10. You perform a database backup.

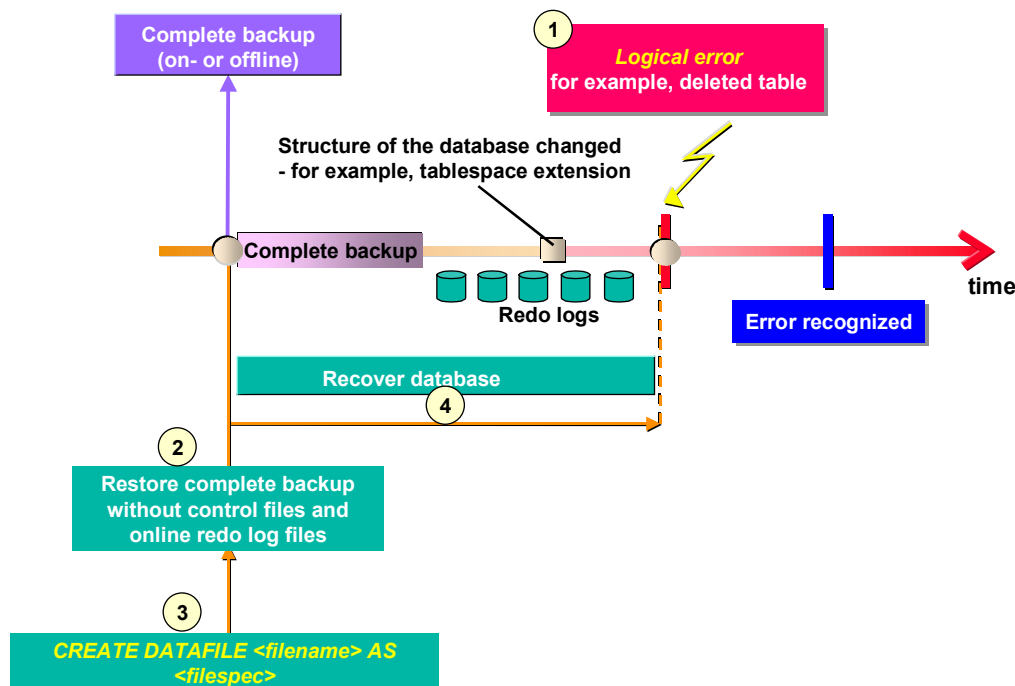
Example

Here are two typical scenarios in which you can use full restore and PIT recovery:

Logical Error



Logical Error with Preceding Structure Change



21.4 Database Reset

Use

This section tells you about resetting your Oracle database after a failure. You normally need to do this if either of the following applies:

- An error occurred in which all copies of the redo log files are lost, but the database file backup is available.
- You performed a full offline backup **immediately** before the error, such as during a software upgrade.

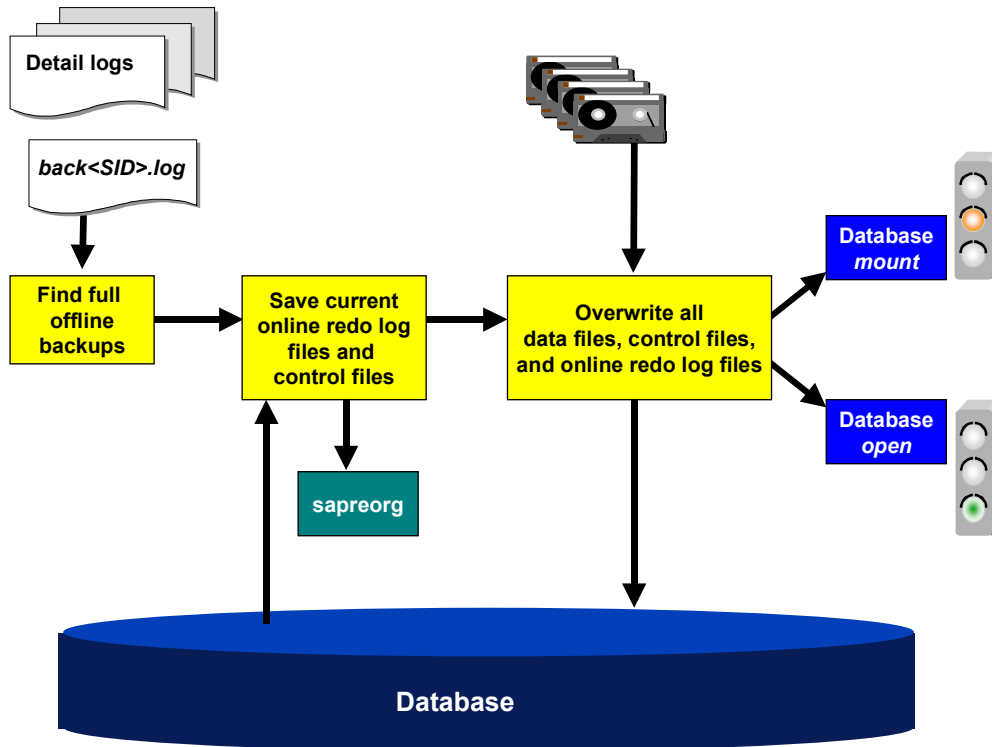
With this function you can reset the database to a previous consistent state, at the time of either a **full offline** or a **consistent online backup**. If you reset from a full online backup, the consistent end point of the backup is used.

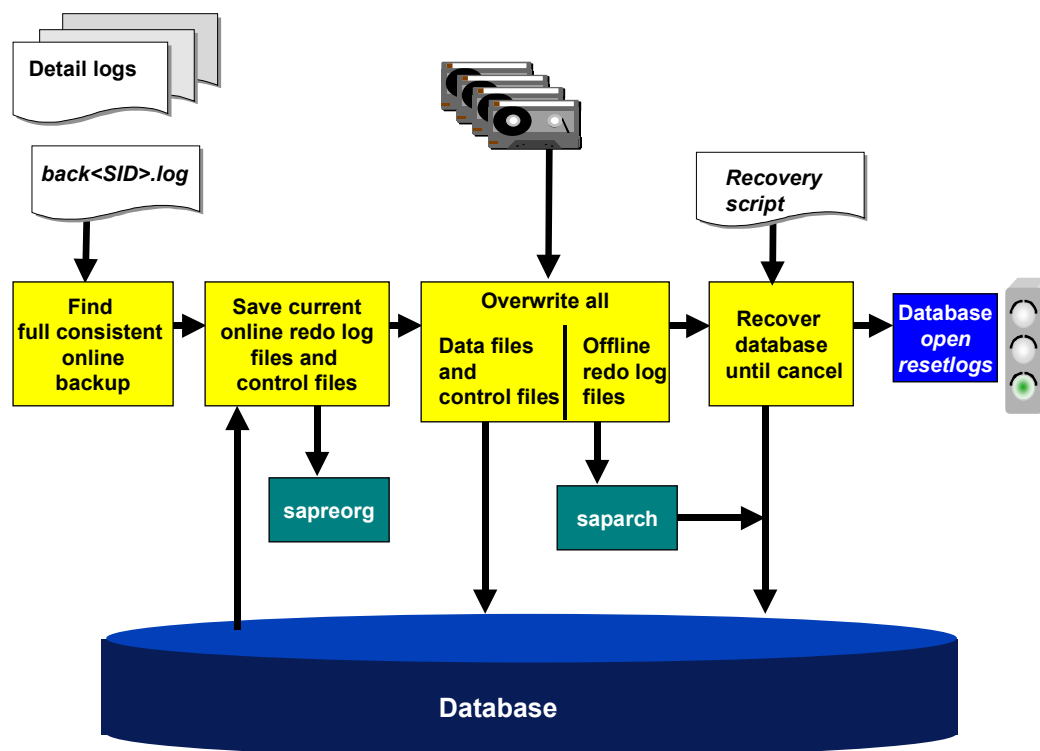
Prerequisites

- We recommend you to:
 - Perform a [full offline backup \[Page 81\]](#). If the database is running, use SAP tools, otherwise use operating system tools.
 - Back up all offline redo log files using BRARCHIVE. For more information, see [-a- archive \[Page 385\]](#).
- It must be possible to set the database to status mount or open.
- You have the following data available:
 - The [BRBACKUP logs \[Page 377\]](#) and the [BRARCHIVE logs \[Page 392\]](#)
 - The data file backups

21.4 Database Reset

- If restoring from a consistent online backup, the accompanying redo log files

Features**Database Reset Using Full Offline Backup****Database Reset Using Consistent Online Backup**



Activities

Database Reset Using Full Offline Backup

1. SAPDBA uses the BRBACKUP summary log file `back<SID>.log` and the corresponding detail logs to find the full offline backups.
2. SAPDBA saves the current online redo log files and the control files to the `sapreorg` directory. This is because these files are overwritten by the database reset. The database must be in the *mount* status for this step.
3. SAPDBA calls BRRESTORE to reset the database, overwriting the data files, control files, and online redo log files with the corresponding files from the offline backup.
4. SAPDBA restarts the database to status mount or open, depending on the option you selected at the start of the reset.

Database Reset Using Consistent Online Backup

1. SAPDBA uses the BRBACKUP summary log file `back<SID>.log` and the corresponding detail logs to find the consistent online backups.
2. SAPDBA saves the current online redo log files and the control files to the `sapreorg` directory. This is because these files are overwritten by the database reset. The database must be in the *mount* status for this step.
3. SAPDBA calls BRRESTORE to reset the database, overwriting the data files, control files, and online redo log files with the corresponding files from the online backup.
4. SAPDBA recovers the database until the consistent end point of the online backup. You cannot specify another point in time for this type of recovery.
5. SAPDBA restarts the database to status open, using the `resetlogs` option.

21.4 Database Reset

6. You must perform a full backup, since recovery after *resetlogs* is not possible.
7. You manually delete the offline redo log files that have been restored from the `saparch` directory.



After a successful database reset with SAP tools, you might need to do some reworking, because the log tables SDBAH and SDBAD are affected. For more information, see [Actions after a Partial Recovery \[Page 504\]](#).

22 Oracle Recovery Manager

Use

The Oracle Recovery Manager (RMAN) is an Oracle backup program. You can use it as a command line interface (CLI) or as a graphical user interface (GUI) in the Oracle Enterprise Manager (OEM). We support RMAN with the SAP backup tools [BRBACKUP \[Page 360\]](#) and [BRARCHIVE \[Page 383\]](#).

RMAN uses the System Backup to Tape (SBT) interface to back up to tape devices. We implement SBT using the SAP backup library. External backup tools can implement this interface as a dynamic link library (DLL).

Integration

By integrating RMAN into BRBACKUP, you can add 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 restore, the control file is restored first, followed by the data files.
- The integration of RMAN into BRBACKUP also guarantees integration into the SAP [Computing Center Management System \(CCMS\) \[Ext.\]](#).
- BRBACKUP tape management functions as previously (that is, as when using the SAP backup library).
- You can still use the BACKINT interface with external tools.
- The SAPDBA user interface remains unchanged, except for some new options.
- All previous SAP backup strategies are supported while using RMAN. Nevertheless, RMAN is not supported for standby database backups and split-mirror backups.
- The following components are delivered with the standard Oracle8 installation:
 - RMAN with the Oracle SBT interface
 - Backup library and backup tool Networker from Legato
 - Legato's BACKINT interface implementation (as of Oracle 8.0.5)

The SAP installation also delivers the SAP backup library with BRBACKUP and BRARCHIVE.

- You can integrate RMAN into your Oracle database with the SAP System using one of the following:
 - SAP implementation

You can use the SAP implementation of the Oracle interface SBT, using the SAP backup library. For more information on activating the SAP backup library, see SAP Note 142635.
 - Legato implementation

You can 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.

Prerequisites

22.1 RMAN Backup Strategies

You are subject to the following restrictions when you use RMAN directly, that is, using the CLI or GUI in the OEM:

- RMAN places information about backups in a recovery catalog. For security reasons, this catalog is held in a separate database on a separate host. This means more administrative work.
- Restore or recovery is complicated in a disaster situation in which you lose your 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.

Features

- Incremental backups
This allows you to change your previous backup strategy and considerably reduce the amount of data to be backed up.
- Consistent backups
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 `DBVERIFY`.
- Reduction in amount of data backed up
Any database blocks that have never been used are not backed up.
- [Verify of backups \[Page 150\]](#)
You can verify backups to tape with the RMAN `VALIDATE` command.
- Reduction in redo log information
The command `BEGIN/ END BACKUP` is not needed in online backups, since the blocks are checked to see if the data is consistent.

22.1 RMAN Backup Strategies

Use


The decision on whether to use the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#) and with which backup library, 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
- Your security requirements

Features

The following table summarizes the main features of backup with RMAN:

	Normal backups	Full backup (level 0)	Incremental backup (level 1)
--	----------------	--------------------------	---------------------------------

With RMAN	<ul style="list-style-type: none"> To tape device with the SAP Backup Library [Page 142] To local disk without backup library [Page 146] To appropriate media with external backup library [Page 144] 	<ul style="list-style-type: none"> To tape device with the SAP Backup Library [Page 142] To local disk without backup library [Page 146] To appropriate media with external backup library [Page 144] 	<ul style="list-style-type: none"> To tape device or remote disk with the SAP Backup Library [Page 142] To local disk without backup library [Page 146] To appropriate media with external backup library [Page 144]
Without RMAN	<ul style="list-style-type: none"> To local or remote tape devices, local or remote disks with BRBACKUP Backup with external tool and BACKINT  <p>You can not use RMAN to make a normal backup to a remote disk [Page 105], even with a backup library.</p>	<ul style="list-style-type: none"> To local or remote tape devices, local or remote disks with BRBACKUP followed by cataloging Backup with external tool and BACKINT followed by cataloging 	Not possible



For more information on the backup types referred to above, see:

- Normal and full backup – see [Complete Backup \[Page 83\]](#)
- [Incremental backup \[Page 83\]](#)

You can also [back up the offline redo log files with RMAN \[Page 149\]](#).

Incremental Backup Strategies with RMAN

You can perform incremental backups of the database with RMAN and BRBACKUP. RMAN offers the following types of incremental backup:

- Several levels
- Cumulative
- Non-cumulative

The integration of RMAN into BRBACKUP is restricted to cumulative incremental backups at level 1. This means that, in the event of a recovery, only one incremental backup at the most has to be applied. In contrast to a full backup, an incremental backup only backs up the **changes** that have been made since the last full backup. This significantly reduces the amount of backup data. Therefore, we recommend this strategy particularly for large databases.

An incremental backup always requires an earlier full backup, that is, a level-0 backup. When RMAN backs up the database, it backs up all Oracle database blocks that have already been used. A subsequent incremental level-1 database backup backs up all Oracle database blocks that have changed since the last full backup. Changes to the whole database are taken into account.

22.2 RMAN Incremental Backups After Structural Changes

Incremental backups at level 1 can only be made with RMAN. However, a full backup can also be made with BRBACKUP or with an external backup tool using BACKINT. In this case, BRBACKUP automatically catalogs the backup as a level-0 backup.



You might not see a reduction in backup time by making an incremental backup. The reason is that it might take as much time to check whether a block has been used or changed as to simply back it up. This means that you only see a significant time reduction when the relatively low throughput of the tape devices is the reason for a long backup.



Using the incremental backup strategy, we recommend you to set a lock period for the tapes of at least 28 days, so that several generations of full backups are available. The offline redo log files must cover the period up to the oldest full backup – that is, both sets of tapes must have the same lock period – and must also be backed up daily.

After structural changes to the database, you can make incremental backups with RMAN and BRBACKUP. This is not possible with RMAN on its own. For more information, see [RMAN Incremental Backups After Structural Changes \[Page 140\]](#).

Activities

This is 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 your specific requirements for backup media, volume of data, and so on.

For more information on incremental backups, see the Oracle documentation.

22.2 RMAN Incremental Backups After Structural Changes

Use

By integrating the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#) into the SAP backup program BRBACKUP – with the [SAP backup library \[Page 142\]](#) or an [external backup library \[Page 144\]](#) – you can make incremental backups, even if the structure of the database has been changed. An example of a structure change is when a data file is added. This is not possible using RMAN on its own.

Features

Save sets are created each time a System Backup to Tape (SBT) backup library is used. Save sets are units that are created on the backup medium. Each save set contains one or more files. During incremental backups, the changes are normally backed up in a single save set. If new files were added to the database since the last full backup, a second save set is created containing the backups of the new files. For more information, see [RMAN Save-Set Grouping \[Page 151\]](#). This means that you do not need to make a full backup of the entire database immediately after database extensions.

In subsequent incremental backups during the cycle, any changes that are made to files that existed when the last full backup was made are saved in one save set. Any new files are fully saved (that is, not just the changes) in the other save set. This method also has the

advantage that you can be more specific about what you want to restore when [restoring an incremental backup \[Page 141\]](#).

22.3 RMAN Restore of Incremental Backups

Use

If you perform an Oracle database backup with the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#), you must generally also restore it with RMAN. However, this does not apply to backups made to disk. SAPDBA supports the following:

- Database reset to an [incremental backup \[Page 83\]](#)
- Partial restore followed by a recovery using incremental backups.

If required, you can continue to recover the database by importing the offline redo log files.

Prerequisites

In contrast to the BRRESTORE restore process (without RMAN) where the database has to be closed, the database must be mounted for an RMAN restore.

Activities

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:

Prerequisite: The database is closed.

1. You restore the control file, and if needed, the online redo log files from the last incremental backup, using BRRESTORE (that is, not using RMAN).

Prerequisite: The database is mounted.

2. You perform a:
 - a. Restore of the full backup (level 0) with RMAN
 - b. Restore of the last incremental backup with RMAN
3. You restore offline redo log files with BRRESTORE.

Prerequisite: The database is mounted.

4. You apply the offline redo log files to the database to reset the data to the required point in time.

If only some of the files are corrupt, 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 importing 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. Be sure to start a full database backup immediately after the database has been successfully recovered.

See also:

[RMAN Incremental Backups After Structural Changes \[Page 140\]](#)

22.4 RMAN Backup with the SAP Backup Library

Use

This section describes how you can use the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#) with the SAP backup library.

Integration

The SAP backup library is an implementation of the Oracle interface System Backup to Tape (SBT) in the form of a Dynamic Link Library (DLL). The Oracle server process calls this DLL to back up data, usually to tape.

Without this library, the Oracle Server process can only back up to a local disk. For tape management reasons, an SBT backup library is always required for a backup to tape.

Features

Backup to Tape Devices

BRBACKUP or BRARCHIVE calls RMAN for a backup to tape devices.

You can use the following profile parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#) for backup to:

- Local tape devices

```
backup_dev_type = tape|tape_auto|tape_box
```

```
tape_copy_cmd = rman|rman_dd
```

- Remote tape devices

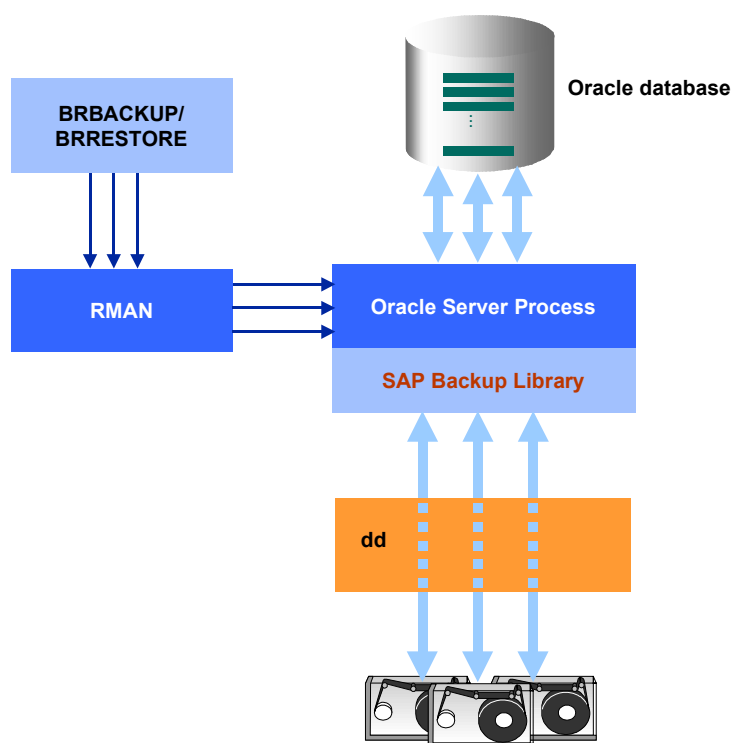
```
backup_dev_type = pipe|pipe_auto|pipe_box
```

```
tape_copy_cmd = rman|rman_dd
```

For more information, see [backup_dev_type \[Page 447\]](#) and [tape_copy_cmd \[Page 477\]](#).

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, as shown in the following graphic:

22.4 RMAN Backup with the SAP Backup Library



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 you to back up a production system in this way.

You can use RMAN to back up to tape as follows:

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

Backup to Remote Disks

As of Release 4.5A, you can make BRBACKUP backups to remote disks. You can use the following profile parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

```
backup_mode = incr
```

```
backup_dev_type = stage
```

```
remote_user = "<user_name> [<password>]" (precondition for SAPFTP)
```

22.5 RMAN Backup with an External Backup Library

`remote_host = <host_name>` (precondition for SAPFTP)

For more information, see [backup_mode \[Page 449\]](#), [backup_dev_type \[Page 447\]](#), [remote_user \[Page 464\]](#), [remote_host \[Page 464\]](#).

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 as follows:

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 program possible

Backups to remote disks are particularly useful for:

- [Standby databases \[Page 162\]](#), when you can use BRARCHIVE to copy offline redo log files to the backup host
- When you use virtual disks to access tape jukeboxes

For more information, see [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).

22.5 RMAN Backup with an External Backup Library

Use

You can use external backup software together with the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#) to back up the database. To back up data to tape, the Oracle System Backup to Tape (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.



Before SAP Web Application Server (SAP Web AS) 6.20, you had to use BACKINT (that is, `backup_dev_type = rman_util`) when backing up with an external backup library. Starting from SAP Web AS Release 6.20, you do not have to use BACKINT.

RMAN saves the database files, but copies of profiles, logs, and a copy of the control files are made to a local or remote disk. Therefore, there are two new values for `backup_dev_type`:

- `backup_dev_type = rman_disk`
- `backup_dev_type = rman_stage`

22.5 RMAN Backup with an External Backup Library

When using a remote disk, you can form a common "staging" area for several systems, which simplifies the further backup of the copied profiles and logs.

Activities

The database is backed up in the following phases:

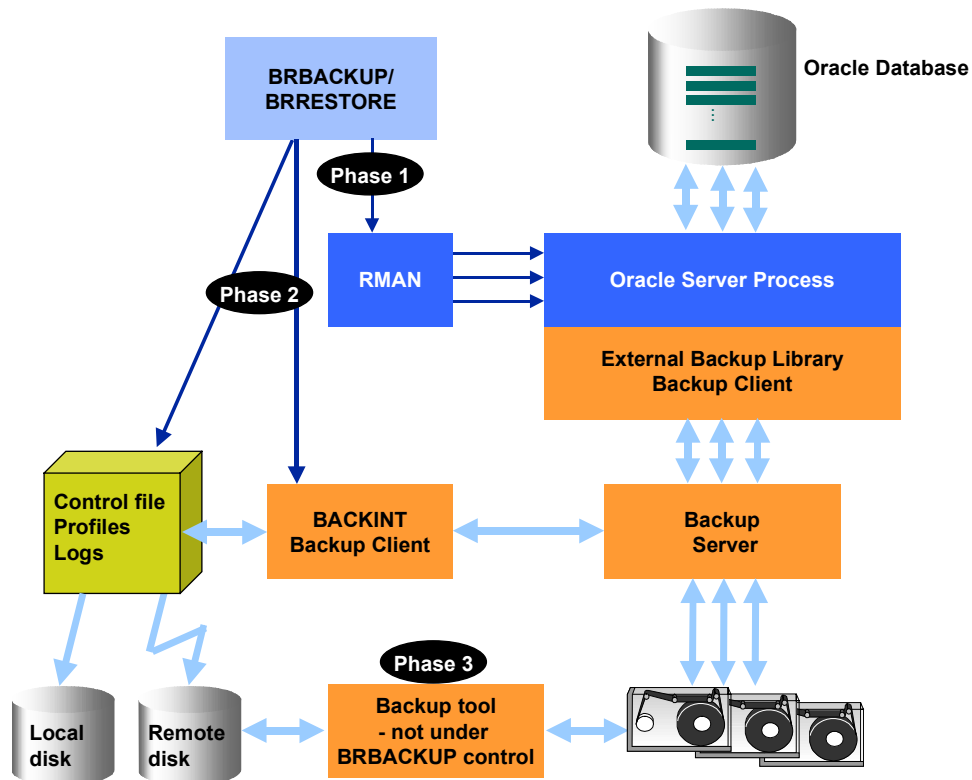
1. BRBACKUP or BRARCHIVE calls RMAN and starts the Oracle server process. The Oracle server process reads the data to be backed up from the Oracle database, and transfers it to the external backup library. The external backup library acts as a backup client. The backup client passes the data to the backup server. The server backs up the data to the storage medium.
2. Normally, 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.
3. You can also let BRBACKUP copy control file, profile, and log files to local or remote disk instead of calling BACKINT. Therefore, you do not need BACKINT any longer when performing RMAN backups with an external backup library.

For more information, see the values `rman_disk` or `rman_stage` for the [backup_dev_type \[Page 447\]](#) parameter.

This feature is available starting with Release 6.20 of SAP Web AS and BRBACKUP 6.10 patch level 6.



You must back up the copies of the control file, profile, and log files to storage medium on your own. You can normally do this directly with the external backup tool.



You can use the following backup variant for external software, with or without RMAN:

22.6 RMAN Incremental Backups Without a Backup Library

<div> <div>backup_dev_type</div> <div>backup_mode</div> </div>	util_file without RMAN	rman_util with RMAN and external SBT backup library	rman_disk rman_stage with RMAN and external SBT backup library
partial	X	X+BACKINT	X
all	X	X+BACKINT	X
full (level 0)	X+cat.	X+BACKINT	X
incr (level 1)		X+BACKINT	X

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, using the following entry in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

```
backup_dev_type = util_file|util_file_online
```

For more information, see [backup_dev_type \[Page 447\]](#) and [backup_mode \[Page 449\]](#).

Subsequent cataloging means that a full backup can be used as a reference backup in an incremental backup strategy.

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 or to local or remote disk.

For more information, see [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).

22.6 RMAN Incremental Backups Without a Backup Library

Use

If you do not have a backup library, you can develop a strategy involving an incremental backup of your Oracle database by performing:

1. A full backup at level 0, which you can perform in one of the following ways:
 - To local disk with the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#), then copy it to tape using BRBACKUP
 - To local or remote disk or tape with BRBACKUP or an external tool (that is, without RMAN), then use RMAN to catalog the backup
2. An incremental backup at level 1 to local or remote disk, for which you always have to use RMAN, then copy to tape using BRBACKUP

For more information, see [RMAN Backup Strategies \[Page 138\]](#).

Features

22.6 RMAN Incremental Backups Without a Backup Library

Full Backup at Level 0 to Local Disk with RMAN

You can make a full backup with RMAN and without a backup library only to **local** disk. You then use BRBACKUP to copy the disk backup to tape. This is a [two-phase backup \[Page 109\]](#), that is, first a backup to disk, then a copy to tape.

It has the following 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.

It has the disadvantage that you need to copy the full backup to tape, which is extra work.

Full Backup at Level 0 with BRBACKUP or an External Tool, Without RMAN

You can make the full backup at level 0 **without** RMAN. After you have made the full backup – using BRBACKUP (with `cpio` or `dd`) or BACKINT and an external tool – BRBACKUP calls RMAN to catalog the backup as a level-0 backup.

This has the following advantages:

- There is additional security when you restore or recover the backup. You can recover the database without RMAN because no save sets are formed during the full backup. In this case, the import of the redo log files created since the last backup replaces the restore of the last incremental backup.
- Under certain conditions the backup can be quicker, since RMAN is not involved in the full backup at level 0. The data is backed up directly using `cpio`, `dd`, or BACKINT.

It has the following disadvantages:

- You cannot use RMAN features such as the database block check.
- The commands `BEGIN BACKUP` and `END BACKUP` are used in an online full backup, leading to extra redo log files. This data increases the load on the system and makes an "archiver stuck" error more likely. If you have an "archiver stuck" error, see SAP Note 316642.
- If you make a recovery without RMAN you might have to apply a lot of redo log files since the full backup is usually only made once a week.

Incremental Backup at Level 1 to Disk with RMAN

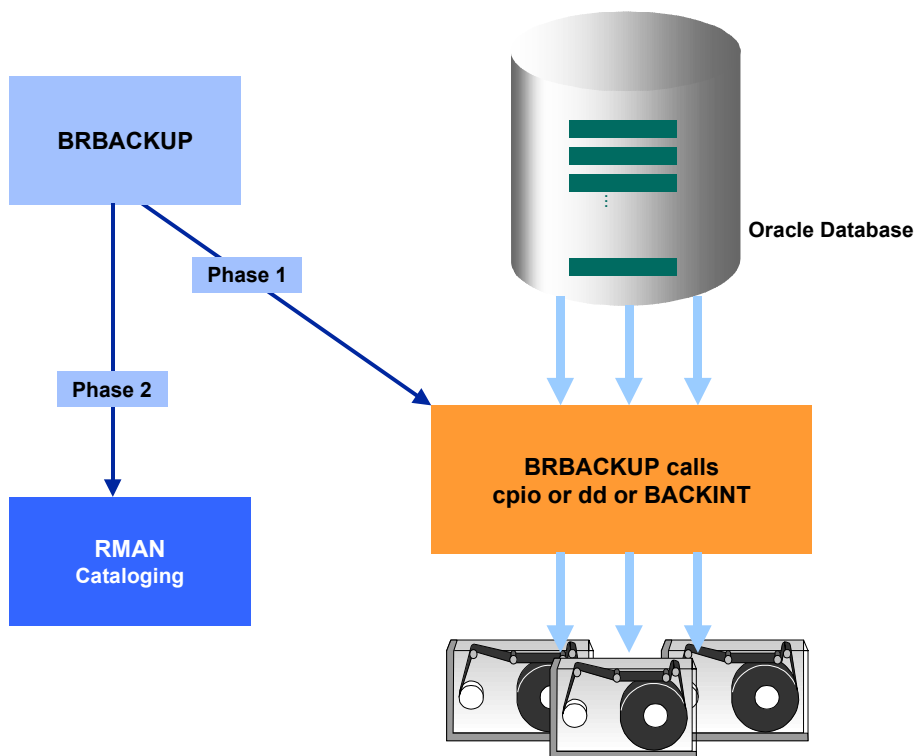
You can make incremental backups at level 1 with RMAN and without a backup library, but only to local disk. There is **no** one-to-one copy of the database files. Instead, save sets are created, which means you can only recover the save sets of an incremental backup with RMAN. For more information, see [RMAN Save-Set Grouping \[Page 151\]](#). In a second phase, you copy the incremental backup to tape with BRBACKUP. This is a [two-phase backup \[Page 109\]](#), that is, first a disk backup, then a copy to tape.

Example

Here is an example of how you can perform the two stages in an incremental backup without backup library: a full backup at level 0 and an incremental backup at level 1.

Stage 1: Full Database Backup at Level 0 Without Backup Library

22.6 RMAN Incremental Backups Without a Backup Library



As shown in the graphic, the full backup consists of the following phases:

1. Phase 1

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 database file names are transferred to external backup software that then backs up the files to the backup medium.

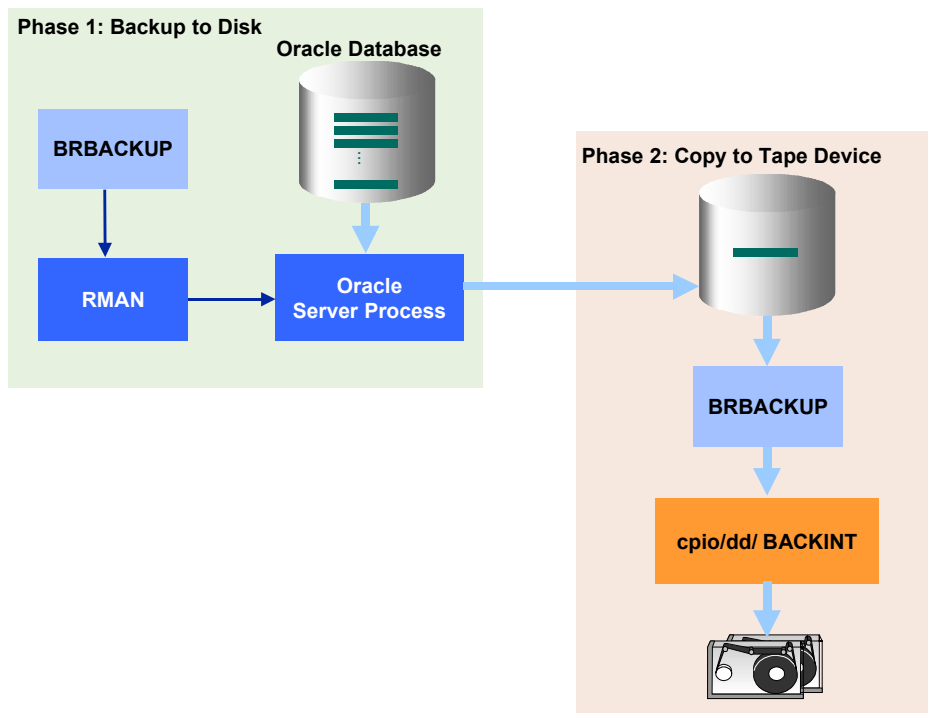
2. Phase 2

BRBACKUP calls RMAN to automatically catalog the backup as a full backup (level 0). This means it can be used as a reference for later incremental backups.

Stage 2: Incremental Backup at Level 1 Without Backup Library

Since you cannot make an incremental backup directly to tape without RMAN, this stage is a [two-phase backup \[Page 109\]](#) (that is, a backup phase followed by a copy phase):

22.7 RMAN Backup of the Offline Redo Log File



As shown in the graphic, the full backup consists of the following phases:

1. Phase 1

BRBACKUP calls RMAN, which internally activates an Oracle server process to read the data from the Oracle database and save it to disk in a save set.

2. Phase 2

BRBACKUP calls cpio, dd, or BACKINT to copy the incremental save set from disk to the backup medium.

For more information, see [RMAN Backup Strategies \[Page 138\]](#) and [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).

22.7 RMAN Backup of the Offline Redo Log File

Use

You can use BRARCHIVE to back up the offline redo log files using RMAN, either with the SAP backup library or with an external backup library. This is available starting with SAP Web Application Server Release 6.10.

The advantage is that the data in the offline redo log files during the RMAN backup is checked for internal consistency. You can think of this as a replacement for the missing verification functionality for offline redo log files when using DBVERIFY.

22.8 RMAN Tape Layout

Definition

The tape layout with RMAN is basically the same as the backup methods without RMAN. This means that you do not have to reinitialize the tapes if you are using the SAP

22.9 RMAN Backup Verify

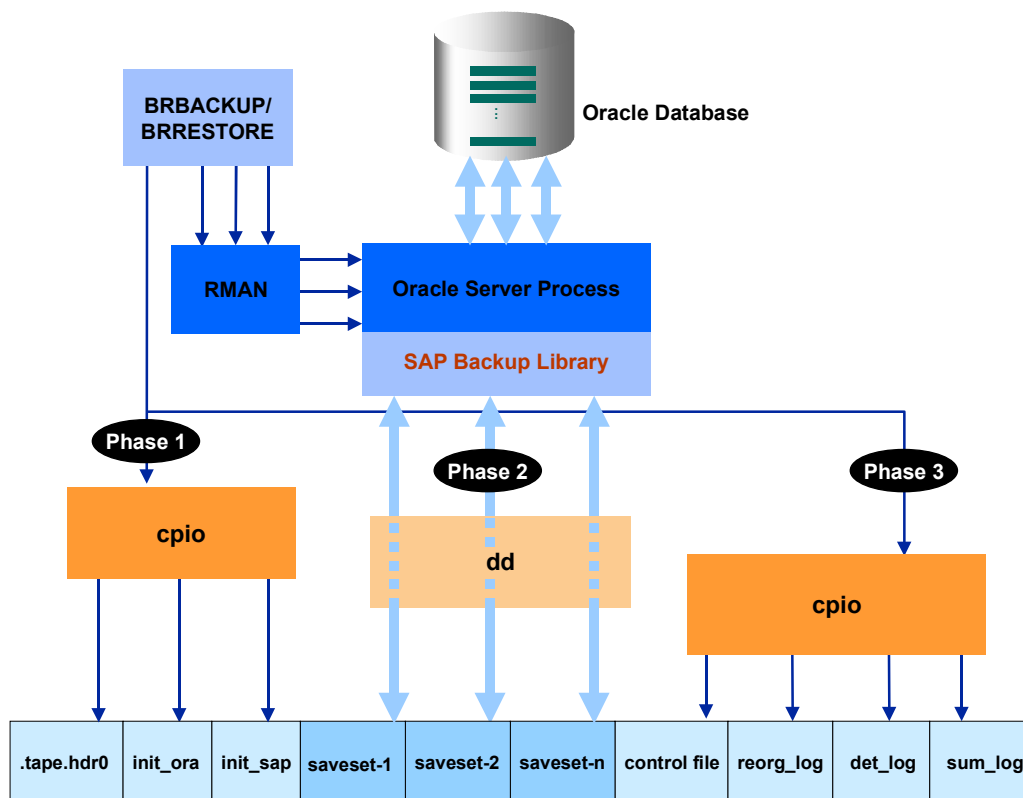
implementation of the System Backup to Tape (SBT) interface – that is, the SAP backup library.

Use

BRBACKUP uses `cpio` to back up the tape administration information (that is, the tape label) and the initialization files `init<DBSID>.ora` and `init<DBSID>.sap` to tape, as follows:

1. The SAP backup library backs up the save sets to tape directly or using `dd`.
2. BRBACKUP uses `cpio` or `dd` to back up the control file and the log files to tape.

Tape Layout



22.9 RMAN Backup Verify

Use

The [BRBACKUP \[Page 360\]](#) and [BRRESTORE \[Page 395\]](#) functions for verifying backups are also supported for backups with the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#).

Features

- BRBACKUP can verify online or offline RMAN backups immediately.
- A delayed verify with BRRESTORE can only be performed on the database host, when the database is mounted or open.
- A verify with DBVERIFY is not supported for RMAN backups, since the Oracle block consistency is checked when the backups are made.

Activities

RMAN verifies savesets with the `VALIDATE` command, as follows:

1. RMAN reads the savesets from the backup medium to check their readability.
2. RMAN checks the internal consistency of the data in the savesets using check sums. No binary comparison is made with the originals.



We still recommend that you verify backups as often as before, that is, at least once in each backup cycle and every week if possible.

For more information, see [Backup Verify \[Page 118\]](#) and [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).

22.10 RMAN Save-Set Grouping

Use

When you back up the Oracle database with the [Oracle Recovery Manager \(RMAN\) \[Page 137\]](#), the SAP backup library helps to optimize the utilization of fast tape drives by combining multiple data files in save sets. Multiple file access – also known as “file multiplexing” – maximizes the flow of data to keep tape drives in streaming mode.

Prerequisites

You can define the number of files contained in each save set with the [saveset_members \[Page 469\]](#) parameter from the [initialization Profile init<DBSID>.sap \[Page 445\]](#) or the [BRBACKUP \[Page 360\]](#) command option [-s|-savesets \[Page 374\]](#).

For a preparation run, set the [backup_dev_type \[Page 447\]](#) parameter to `rman_prep`.

For an incremental backup with the SAP backup library, the [saveset_members \[Page 469\]](#) parameter is internally set to `all` so that only one “incremental save set” is created including all changed blocks.

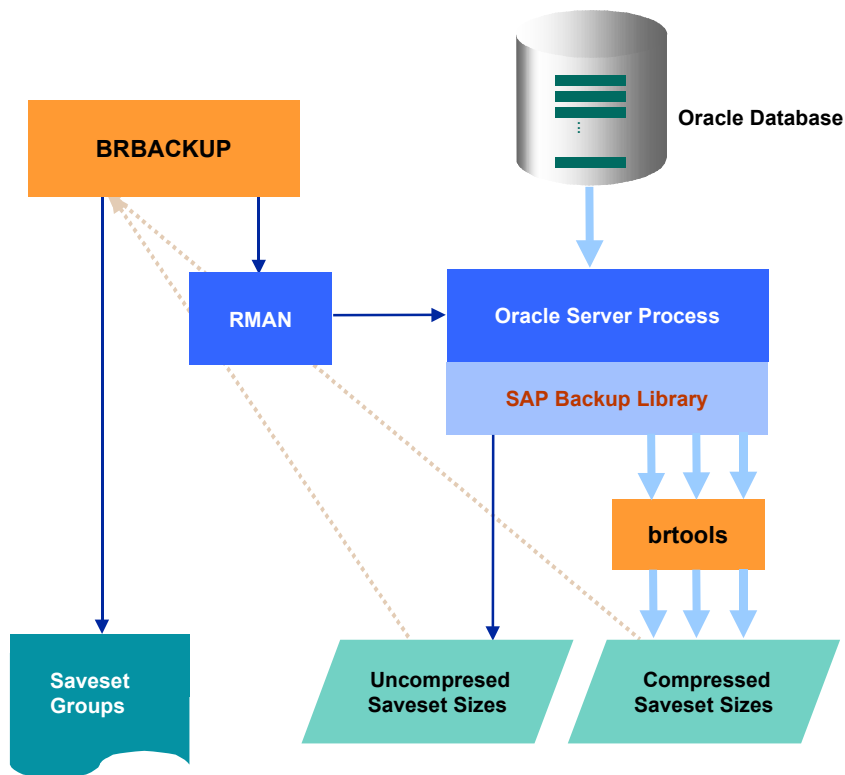
Features

The SAP backup library optimizes the distribution of the data files across the save sets in a preparation run as follows:

1. It estimates the expected compression of the save sets by using BRTOOLS to determine the decompressed and compressed save set sizes.
2. BRBACKUP forms save sets and stores the information on the composition of the save sets and the compression rates for future backups in the database.

The following graphic shows this:

22.10 RMAN Save-Set Grouping



A save set can contain from one to four individual data files, all files of a tablespace, or the complete data backup. The size of the save sets for the backup must be selected according to the tape device. A fast data flow with a minimum save set size is the optimum. Large save sets are not recommended, since in a restore the complete save set has to be read, even if only one data file is required.

Activities

SAP recommends that you perform the preparation run once a month – using the **BRBACKUP** command `-d|-device` [Page 367] with the `rman_prep` parameter – and each time that major changes have been made to the database, for example after a reorganization, an SAP release change, database migration, or mass data import. 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 380].

23 Advanced Backup and Recovery

This section deals with advanced topics in backup and recovery. Make sure that you have sufficient expertise before using these functions.

23.1 Backup of Large Oracle Databases

Use

This section gives you advice on backing up large Oracle databases. Large databases contain 500 GB – 2 TB or more data. For reasons such as the following, you might not be able to back up the database as often as you want to (that is, daily in most cases):

- Performance problems
There is a heavy load on database server resources, particularly CPU time, system and I/O buses, disk and volume controllers. Therefore, online operations in the SAP System are restricted.
- Lack of time
Although you normally perform backups at times of low system load, such as at night, you might still find that you exceed the 12 hours available.
- Data volume
The amount of database data is too great to back up in the time available.
- Common backup strategy
A common backup strategy for database and non-database files is not a solution since non-database data can better be backed up with operating system tools (such as `tar`).

Prerequisites

Your system is normally configured so that there is a single server for the database, where no other large applications run.

We recommend you **not** to back up large databases across a network because of instability and performance problems.

Features

Whether you can back up large databases with [BRBACKUP \[Page 360\]](#) (and `cpio` or `dd` 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 buses
- Maximum throughput of the system bus
- `cpio` and `dd` performance, determined by internal buffering and blocking. In general, `dd` offers much better performance than `cpio`

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 might require multiple tape changes and the management of hundreds of

23.1 Backup of Large Oracle Databases

volumes. If possible, use tape jukeboxes or robots supported by the BACKINT interface and external backup programs.

You can reduce the amount of data significantly by using the following features:

- [Incremental Backup \[Page 83\]](#), which we recommend
- [Split-Mirror Disk Backup \[Page 170\]](#)
- [Standby Database \[Page 162\]](#)

23.1.1 Backup Devices for Large Databases

Definition

This section discusses how to use devices when [backing up a large Oracle database \[Page 153\]](#).

Use

- Do not perform the backup 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 \[Page 360\]](#) is 255.
 - Backup devices (for example magneto-optical media) that are addressed with an external backup program can be reached using the BACKINT interface. For more information, see [External Backup Programs for Large Databases \[Page 155\]](#).
- BRBACKUP offers only limited support for [automatic tape changers \[Page 176\]](#) such as jukeboxes. However, you can address such devices using the BACKINT interface to external backup programs.
- Use tape units with a larger capacity and higher throughput rate if possible.
- Factors other than the performance of the backup device also play a large role, such as:
 - Server throughput, in particular, hard disk access times, system bus speed, I/O bus speed.
 - 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, the load is split between the mounted devices.

23.1.2 Backup of Large Databases to Tape with BRBACKUP

Use

This section describes how you can use [BRBACKUP \[Page 360\]](#) to [back up large Oracle databases \[Page 153\]](#) to tape. BRBACKUP calls cpio or dd to copy individual database files from disk to tape. As a result, throughput is largely determined by cpio or dd.

Prerequisites

Note the following restrictions for backup with BRBACKUP:

23.1 Backup of Large Oracle Databases

- As the database server is working with high load during the backup, we recommend you to reduce other activities on the computer to the absolute minimum during the backup. Virtually all of the computer's resources ought to be available for the backup.
This clearly limits the high availability required of the SAP System. However, there is no simple solution to this conflict.
- Be aware of the features of [hardware compression \[Page 101\]](#) if you decide to use this.
- No cpio continuation mechanism is supported during parallel backups. For more information, see [cpio Continuation Tape \[Page 359\]](#). This means that you must always completely save individual files to one volume. Always perform tape swapping with 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 – that is, label, `init<DBSID>.ora` file, and `init<DBSID>.sap` file – and the log files at the end of the tape (central, detailed, and summary log) are not taken into account.

Features

BRBACKUP offers the following functions for optimal backup of large databases:

- BRBACKUP can back up in [parallel \[Page 113\]](#) to several locally mounted tape units (up to 255).
- In a parallel backup, BRBACKUP also supports [automatic tape changers \[Page 176\]](#). Therefore, a completely automatic backup is also possible with tape swapping for one or more tape units during a backup. For more information, see [Unattended Backup \[Page 114\]](#).
- All files to be saved are distributed to the tape volumes inserted in the tape units. BRBACKUP has different optimization targets. For more information, see [Optimization of File Distribution \[Page 156\]](#).

See also:

[Optimization of File Distribution \[Page 156\]](#)

[Optimization when Using a Logical Volume Manager \[Page 158\]](#)

[Partial Backups \[Page 159\]](#)

23.1.3 External Backup Programs for Large Databases

Use

You might want to use external backup programs if you need to [back up a large Oracle database \[Page 153\]](#). We provide the BACKINT interface for this purpose. For more information, see [External Backup Programs \[Page 180\]](#).

Features

The main advantage of using the BACKINT interface with external backup programs is that you can use other backup media, for example, magneto-optical (MO) media. Alternatively you can transfer volume management to other systems, for example, [automatic tape changers \[Page 176\]](#), such as jukeboxes and tape robots.

23.1 Backup of Large Oracle Databases

For more information, see the option `util_file_online` with the [backup_dev_type](#) [Page 447] parameter or the corresponding command option `-d|-device` [Page 367].



You can obtain information on the throughput and performance of external backup programs from the vendors providing the external backup.

23.1.4 Parallel Backup of Large Databases to Disk with BRBACKUP

Use

You can use [BRBACKUP](#) [Page 360] to back up your large Oracle database to [multiple disks](#) [Page 105] in [parallel](#) [Page 113].

Prerequisites

You must define the directories with the [backup_root_dir](#) [Page 450] parameter.

Features

- You can perform the backup with or without [software compression](#) [Page 102].
- You can control the degree of parallelism – that is, the number of parallel copy processes – using the [exec_parallel](#) [Page 459] parameter or the BRBACKUP command option `-e|-execute` [Page 368].
- If required, you can make a [two-phase backup](#) [Page 109], that is, first to disk and then to tape.

Activities

To avoid imposing extra processing load on the database server during the second phase of a two-phase backup, consider the following procedure instead:

1. Unmount the file system from the database server.
2. Mount the file system on a second host.
3. 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 unmount and mount operations – that is, `umount` and `mount` – are necessary, since the file system cannot be mounted on different computers at the same time due to the buffering mechanism.

Alternatively, you can [back up directly to remote disks](#) [Page 105]. This eliminates the need for `umount` and `mount`.

Another approach to reduce processing load on the production database host is to use [split-mirror disk backup](#) [Page 170].

23.1.5 Optimization of File Distribution

Use

This section describes how [BRBACKUP](#) [Page 360] optimizes file distribution during an Oracle backup. This is especially relevant to [backup of large Oracle databases](#) [Page 153].

Features

23.1 Backup of Large Oracle Databases

When distributing the files to the tapes, [BRBACKUP \[Page 360\]](#) has different optimization targets:

- All files from a disk saved to the same tape volume

Therefore, try to avoid competing disk accesses, and keep the number of read/write accesses to a minimum.

If a Logical Volume Manager (LVM) is used, this target can only be attained if the logical volumes are not scattered over several physical hard disks. For more information, see [Optimization with a Logical Volume Manager \[Page 158\]](#).

- Time-based optimization used

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. Note the following factors relevant to time-based optimization:

- It 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. Therefore, the optimization is performed from the backup times stored by BRBACKUP.
- The tape capacity in general is not used to its maximum due to time-based optimization. That is, 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. Therefore, the total capacity of all available tapes inserted at the same time must significantly exceed the total size of the files to be saved.
- The time-based optimization is cancelled internally if it would cause a tape change, since 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.

- All mounted tapes used

To minimize the total backup time, BRBACKUP always attempts to use **all** mounted tape units, even if the backup would fit onto a smaller number of tapes.

The one exception is when the number of files to be backed up is less than the number of tape devices.

- Files sorted and distributed by size

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, because the tape capacity can never be exceeded and the total tape capacity can be better utilized. For more information, see the [tape_size \[Page 479\]](#) parameter.

- Previous backup times used

BRBACKUP is capable of “learning.” It stores the backup times of individual files in the database and uses these in the next backup for time-based optimization (if this is performed).

As a result, the changes to backup time are taken into account for individual files. The backup time can change, for example, if the fill level of the individual database files varies and the files are saved with [hardware compression \[Page 101\]](#), using a tape unit.

- Parameter `-o|-output` with options `dist`, `time used`

BRBACKUP or BRARCHIVE use these to:

23.1 Backup of Large Oracle Databases

- Display the file distribution performed by BRBACKUP before starting the actual backup
- Display the backup times of the individual files after the backup is completed

For more information, see [-o|-output \[Page 372\]](#) (BRBACKUP) and [-o|-output \[Page 389\]](#) (BRARCHIVE).

See also [Log Supplements \[Page 484\]](#).

23.1.6 Optimization with a Logical Volume Manager

Use

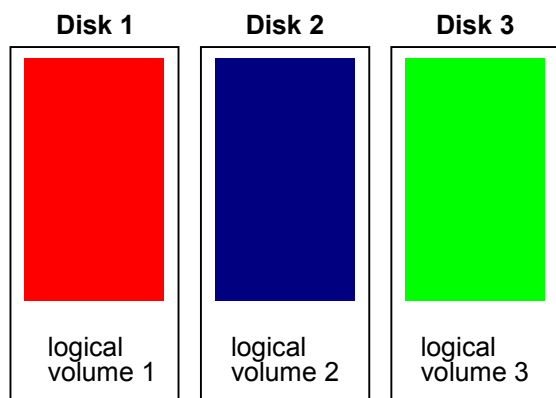
This section describes how a Logical Volume Manager (LVM) influences the [optimization of file distribution \[Page 156\]](#) during an Oracle database backup.

If you use an LVM, BRBACKUP can only save all files from a hard disk to tape if the logical volumes are not scattered over several hard disks. Configuration A of the graphic below is better suited for backup with an LVM.

Features

Configuration A

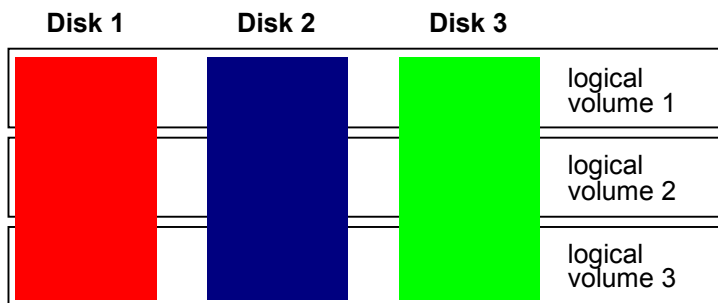
Each logical volume exactly corresponds to one hard disk. For example logical volume 1 = disk 1, logical volume 2 = disk 2, logical volume 3 = disk 3.



Configuration B

Each logical volume is set up so that it covers areas of all the hard disks:

- 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. However, configuration B can provide better performance for online operation of the SAP system.

Activities

When planning the configuration of logical volumes for large databases, you need to find a compromise between the following:

- More effective backup
- Higher SAP System performance

Consider the following factors:

- 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 possibly – in configuration B – reduced optimization with BRBACKUP
- Although the higher security and availability of your datasets provided by using the LVM has a generally high priority, consider whether you could do without an LVM:
 - You lose the LVM advantages outlined above.
 - However, you can perform a dataset backup much more effectively using BRBACKUP.
- In large databases, the configuration of the database files is not variably selectable. You must consider the effect on physical disk configuration and the influence this has on performance when you make changes to the configuration, such as structure changes due to a tablespace extension or a tablespace reorganization with data files.

23.1.7 Partial Backups

Use

You can reduce long backup times for your Oracle database by dividing the backup into several partial backups.

Features

Advantages

- The partial backups require less time, so you can perform them daily.
- You can recover the database at any time if the corresponding redo log entries exist. It is also possible to restore the entire database and then recover it.

23.1 Backup of Large Oracle Databases

- You can quickly recover after a media error – that is, a disk crash – as you only need to restore the files of the affected hard disks.

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, such as program errors or user errors. However, this is an unlikely event.
- You must ensure that **all** database files are saved in a partial backup cycle. This is your responsibility as database administrator.



Back up at [tablespace \[Page 160\]](#) level (not at database file level, which is also possible), as BRBACKUP can then make sure 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, you can put extremely large tables into separate tablespaces, and then back them up separately at tablespace level. Use BRBACKUP command option [-f-fillup \[Page 369\]](#) to do this.

23.1.8 Tablespace Backups

Use

It is important to keep the volume of data to be backed up or restored as small as possible, especially when backing up large Oracle databases. One way to do this is by backing up specific tablespaces.

Activities

You do **not** have to back up pure index tablespaces during a database backup. Index definitions are stored in the `SYSTEM` tablespace as a matter of course. Therefore, indexes can always be recreated.

To make a backup of all tablespaces except index tablespaces, perform one of the following:

- Start BRBACKUP with the command option `-m all_data`.
For more information, see [-m|-mode \[Page 370\]](#).
- Set the profile parameter `backup_mode = all_data` in `init<DBSID>.sap`.
For more information, see [backup_mode \[Page 449\]](#).

During the backup, BRBACKUP indicates all pure index tablespaces, so that BRRESTORE recognizes these. In the restore process, which uses a complete backup (`brrestore -m all`), you can exclude the pure index tablespaces with the command `brrestore -m all_data`. For more information, see [-m|-mode \[Page 403\]](#).



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. SAPDBA automatically performs this during the recovery. For more information, see [Database Restore and Recovery with SAPDBA \[Page 321\]](#).



After new indexes have been created, you must [update optimizer statistics \[Page 420\]](#) on these indexes. For example, you can use `brconnect -f stats -t <tablespace name>` to do this. For more information, see [_f stats \[Page 436\]](#). This requires extra time, possibly canceling the expected time reduction from a new index. We recommend you to test the **complete** recovery procedure to find out whether you can indeed expect a time reduction of this approach.

23.1.9 Backup Example for a Large Database

The following example shows how you can perform a large Oracle database backup:

- Size of the database:
The size of the database is 400 GB.
- Backup devices
The database backup is performed to locally mounted tape units. Four tape units are available in our example.
- Throughput
The throughput of the backup is approximately 15 to 20 GB per hour.
- Tape capacity
About 70 GB for tape units with [hardware compression \[Page 101\]](#), assuming the average compression rate for SAP data, about 3-4 GB per hour.
- Duration
The backup should be possible in less than 10 hours overnight.

If the database can be backed up in parallel to four tape devices, then it can be completed within ten hours, using hardware compression.



Remember that scalability is restricted. This means that backup performance does **not** rise in proportion to the number of tape devices. Therefore, if you use more tape devices, backup time increases for each device.

23.1.10 Speeding Up the Backup

Use

This section discusses how you can speed up a backup of your Oracle database. Backups can often take a long time. For example, if you use digital audio tape (DAT) devices as a backup medium, you can back up 4 to 6 GB per hour, using [hardware compression \[Page 101\]](#). This means that backing up a 50 GB database on one tape device of this type takes approximately 10 hours.

Features

Parallel Backup

You can improve throughput by performing data backups [in parallel \[Page 156\]](#). 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

23.2 Standby Database

make a backup of, for example, a 100 GB database in parallel on five tape devices in five hours.

For more information, see [Parallel Backup \[Page 113\]](#), [tape_address \[Page 475\]](#), [tape_address_rew \[Page 477\]](#), and [backup_root_dir \[Page 450\]](#).

Logical Volumes

If you are using the [logical volume manager \(LVM\) \[Page 158\]](#), a logical volume is considered to be one disk. Therefore, you should not distribute a logical volume to be used for backup over several physical disks when this is not absolutely essential.

By keeping the logical volume on one physical disk, you minimize read/write head movement on the disk, so speeding up the backup.

Activities

A good way to reduce backup time is to perform [two-phase backup \[Page 109\]](#):

1. You back up to disk, using the [backup_dev_type \[Page 447\]](#) parameter.
2. You copy the disk backup to a tape volume, using the [tape_copy_cmd \[Page 477\]](#). You can assign the copy operation a lower priority.

23.2 Standby Database

Use

The standby database is supported officially by Oracle as of Version 7.3. The Oracle documentation contains detailed information on this database configuration. For more information, see [Standby Database Configuration \[Page 163\]](#).



Only use the standby database if you are an experienced user. If you decide to use the standby database, **you are fully responsible** for correctly configuring and running the standby database.

Integration

The SAP tools [BRARCHIVE \[Page 383\]](#) (for backup of offline redo log files) and [BRBACKUP \[Page 360\]](#) (for database backup) support the standby database. For more information, see:

- [Standby Database: BRARCHIVE Backup of Offline Redo Log Files \[Page 165\]](#)
- [Standby Database: BRBACKUP Backup of Database Files \[Page 167\]](#)

BRARCHIVE and BRBACKUP support the standby database by helping you to:

- Copy the offline redo log files from the production database to the standby database, including a check on the copied files
- Import the offline redo log files into the standby database (recovery), followed by backup of the offline redo log files to tape
- Back up the standby database
- Create and configure a new standby database, and reconstruct the production database after a takeover

Features

Before you use the standby database, weigh up the advantages and disadvantages carefully.

Advantages

- Very low failure rate

All system components are duplicated. The primary and standby instances can run on different hosts. They can also have separate locations depending on the safety requirements.

- Very short downtimes

If an error occurs in the primary database system and it is necessary to recover the database, you can perform the recovery very quickly on the standby host. This avoids a time-consuming data-file restore, since these files are already located on the standby host. The only thing you need to do is to import the last entries from the redo log files. Therefore, the standby instance can take over the tasks of the primary instance very quickly.

- Significant decrease of the load on the production host

The database backup requires considerable resources and time for large databases. Since the backup can run on the standby host, the load on the primary instance is reduced significantly. Therefore, the resources on the production host are fully available for production operation, and database operation does not need to be interrupted or restricted for a backup.

Disadvantages

- High costs

For a standby database scenario, all system components must be available in duplicate. In particular, duplicate hardware resources (CPU, hard disks, and so on) are expensive.

- High system administration expense

You need to set up the standby host. If structural changes are made on the primary database system, make sure these are incorporated on the standby host. When the standby instance has taken over production operation – a “takeover” – you must set up a replacement standby database.

- High requirements for switchover software

For the standby instance to take over production operation, the appropriate switchover software is required. You need to work with the hardware and software suppliers, who are responsible for selecting this software and ensuring that it functions correctly.

23.2.1 Standby Database Configuration

Use

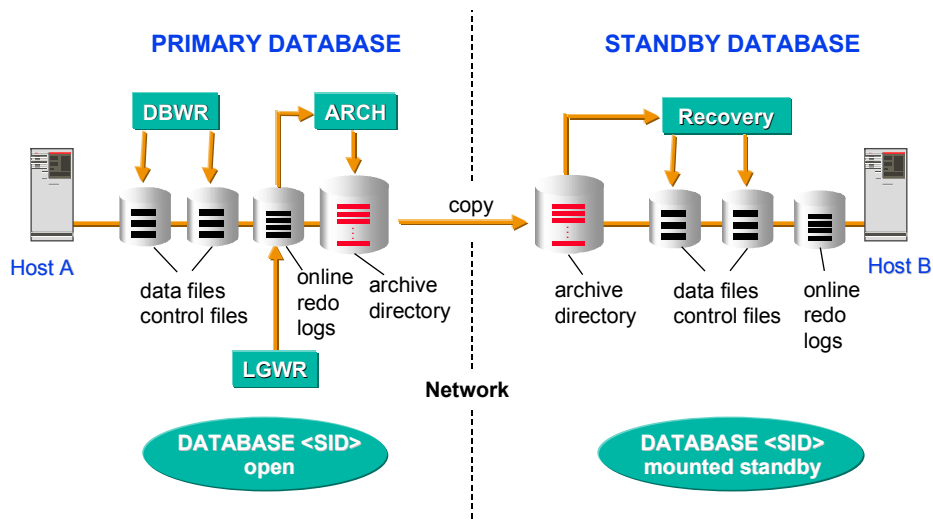
This section describes how to configure the Oracle Standby Database. For more information, including advantages and disadvantages of using the standby database, see [Standby Database \[Page 162\]](#).

When the primary (that is, production) 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.

Features

The following graphic illustrates the standby database:

23.2 Standby Database



Two identically configured databases operate on two identically configured hosts:

- The primary (that is, productive) Oracle instance is located on the first host. The database is open and fully available for all SQL statements of the SAP System. The primary database system is also the system that 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 chosen delay. To do this, the offline redo log files created in the primary database system are applied (only the redo entries already archived by Oracle can be imported).

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 a very short time. This is a "takeover," which means that the recovery mode of the standby instance is terminated and the standby database is 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 might 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, do **not** copy the current online redo log file to the standby host. If you try to directly apply the redo entries from the online redo log file, you might crash the standby database. 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. You might 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 you need to intervene manually to incorporate the structural change in the standby database. After that, you need to restart the recovery process.

The original names of the primary database ought to be retained. From SAP Release 4.5B, BRBACKUP supports renaming all database files to another `SAPDATA_HOME` directory of BRBACKUP using the `init<DBSID>.sap` parameter [orig_db_home \[Page 463\]](#). If you use this parameter, it is even possible to run the standby database on the same host as the primary database. However, we do **not** recommend this for high availability.



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. However, they are recorded in the standby database `ALERT` file. Therefore, be sure to check the `ALERT` file regularly.



For more information, see the Oracle documentation. The new and changed SQL and SQLPLUS commands are also described there as well as the necessary `init.ora` parameters, which are required for working with a standby database.

Activities

To create a standby configuration, you make a copy of the production database. You can then use this as the standby database, as follows:

```
backup_dev_type = disk_standby|stage_standby
```

For more information, see [backup_dev_type \[Page 447\]](#).



23.2.2 Standby Database: BRARCHIVE Backup of Offline Redo Log Files

Use

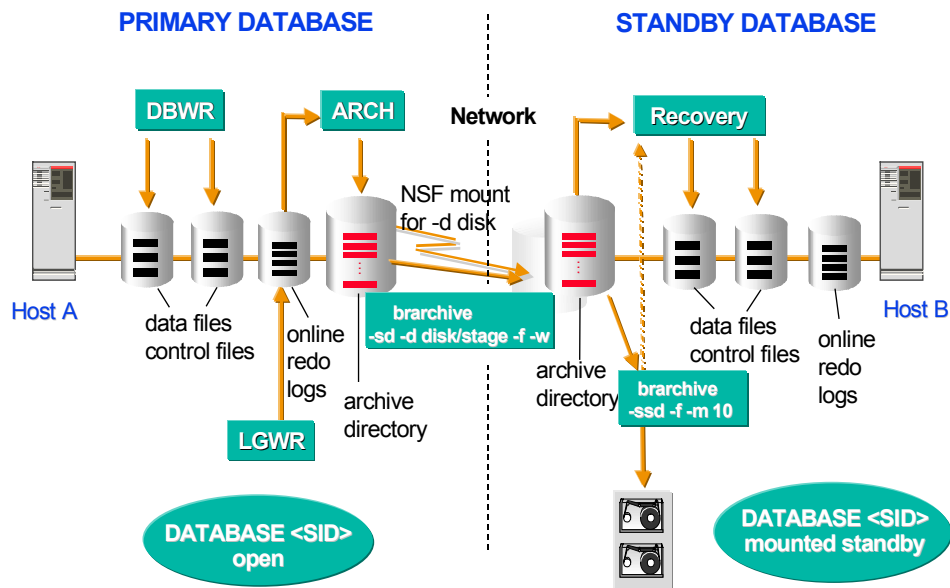
You can use [BRARCHIVE \[Page 383\]](#) with the Oracle [Standby Database \[Page 162\]](#) to back up the offline redo log files.

BRARCHIVE can back up offline redo log files from the primary to the standby instance. This is possible because BRARCHIVE is able to copy offline redo log files to a local or remote hard disk.

Features

The following graphic illustrates how BRARCHIVE works with the standby database:

23.2 Standby Database



- **BRARCHIVE process on the primary host**
This process copies the offline redo log files to an NFS-mounted or remote directory, which represents the archive directory (usually `saparch`) on the standby host. For a mounted directory, the copy process uses an insecure network protocol. In this case, you should use BRARCHIVE with the `-w|-verify` [Page 391] option.
- **Reliable copy programs**
Starting with SAP Release 4.5A, BRARCHIVE can use reliable copy programs such as `rsh` or `ftp` for the copying of the offline redo log files from the primary host onto the standby host, with the `backup_dev_type` [Page 447] parameter set to `stage`. Therefore, NFS-mounting and verification are no longer necessary.
- **BRARCHIVE process on the standby host**
This process waits for the offline redo log files to be copied into the local archive directory. If a redo log file was copied completely, BRARCHIVE applies these redo entries to the standby instance with the `-m|-modify` [Page 389] option, backing up the redo log file and deleting it if necessary. Therefore, BRARCHIVE starts the recovery process of the standby database, in which the offline redo log files are processed individually.
- **Delaying import of redo entries**
You can delay importing the redo entries, using the `<delay>` parameter with the `-m|-modify` [Page 389] option. If a logical error occurs in the primary instance (for example, accidental deletion of a table), you can prevent this error from being imported in the standby instance, by stopping the BRARCHIVE run.
- **Importing offline redo log files**

You import the offline redo log files with the following Oracle command:

```
RECOVER STANDBY DATABASE;
```

Starting with BRARCHIVE Release 4.5B, when the volume of offline redo log files is high, the import to the standby database is performed in parallel to the backup, leading to significantly faster processing. For more information, see `-m|-modify` [Page 389].



To import the redo log files, the database user (usually SYSTEM) must have SYSDBA authorization.



When redo entries are imported in which a structural change of the primary database is recorded, the BRARCHIVE process terminates 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>'

You now need to manually incorporate the structural change in the standby database. To do this, you can use the command:

```
ALTER DATABASE CREATE DATAFILE '<file_name>';
```

You can now restart BRARCHIVE.

For more information, see SAP Note 216108.

23.2.3 Standby Database: BRBACKUP Backup of Database Files

Use

You can use [BRBACKUP \[Page 360\]](#) to back up the data files and control files of the [Oracle Standby Database \[Page 162\]](#).

A major advantage of the standby database is that you do not have to perform backups on the primary (that is, production) database. Instead, you can perform the backup on the standby database using BRBACKUP. This means that the database backup does not add to the load on the primary database host. Since there is no online operation on the standby database, all host resources are available for the database backup.

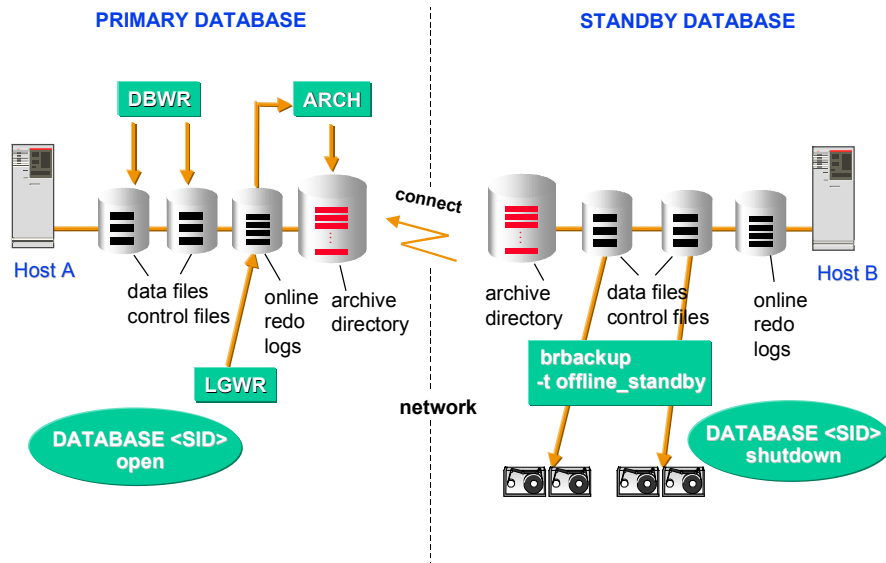
Prerequisites

- The standby instance is in the recovery state and is not opened. You can only perform an offline backup. For the BRBACKUP backup of the standby data, you must set the [backup_type \[Page 450\]](#) parameter to `offline_standby`.
- Renaming of database files in the standby database is **not** supported by BRBACKUP. You must keep the original names of the primary database.
- From SAP Release 4.5, we support renaming all database files to another SAPDATA_HOME directory of BRBACKUP using the [orig_db_home \[Page 463\]](#) parameter. If you use this parameter, you can even run the standby database on the same host as the primary database. However, we do **not** recommend this for high availability.
- For connection to a remote host, you must meet the [remote database connect requirements \[Page 169\]](#).

Features

The following graphic shows how backup occurs in a standby database:

23.2 Standby Database



- BRBACKUP logs on to the primary database instance using the instance string from the [primary_db \[Page 463\]](#) parameter. It retrieves the information required on the database structure. This information is written to the backup logs.
- BRBACKUP stops the standby database instance.
- BRBACKUP backs up the standby data.
- After the backup has finished, the original state of the standby database instance is recovered. If the database was in a recovery state, this state is restored, using Oracle commands `STARTUP NOMOUNT`, `ALTER DATABASE MOUNT STANDBY DATABASE`.

See also:

[Structure-Retaining Database Copy \[Page 111\]](#)

23.2.4 Standby Database: Restore and Recovery

Use

This section describes what you need to do in the event of a failure with the Oracle [Standby Database \[Page 162\]](#).

Activities

If the production database fails, you can do one of the following:

- Switch to the standby database, which now becomes the production database



If you switch to the standby database, be sure to perform backups as described below in "Control File Backup and Complete Online Backup After a Switch."

After you have fixed the problem on the previous production database, you can do one of the following:

- Stay with the swapped configuration

This means that you continue production operation on the previous standby database. In effect, you have swapped the configuration, with the previous production database becoming the standby database.

- Resume the original configuration

This means that you switch back to the previous production database, to resume the configuration from before the failure. For more information, see "Resuming the Original Configuration after a Switch" below.



Which option you choose depends on whether the production and standby databases are equivalent in terms of processing power. Often, the standby database is less powerful, in which case it makes sense to switch back to the original production database as soon as you can.

- Fix the problem and continue on the production database. If necessary, restore and recover the database with the redo log files, if available.

Control File Backup and Complete Online Backup After a Switch

If the production database fails, you can do one of the following. After a takeover, you should back up the control file immediately. Otherwise, you cannot recover the standby database at all. For example, you can enter the following command to do this:

```
brbackup -m 0 -t online
```

For more information, see [-m|-mode \[Page 370\]](#).

We then **strongly recommend** you to perform a complete online backup as soon as possible. For example, you can enter the following command to do this:

```
brbackup -f m all -t online
```

For more information, see [-f|-fillup \[Page 369\]](#).

This is the only way to make sure that the standby database can be recovered, if necessary.

Be sure to perform this operation **whenever** you switch production from one database to another.

Resuming the Original Configuration After a Switch

To recreate and configure a production database after a switch, you use BRBACKUP to perform an offline backup of the current production database (that is, the database that was functioning as a standby before the switch). To do this, set:

```
backup_type = offline_stop
```

For more information, see [backup_type \[Page 450\]](#).

After the backup, you directly switch the database to mount-standby state, so it can resume the role of the standby database. After a restore on the production database, this can resume its role as production database. If you want to set it up in one step (that is, without a restore), you also set one of the following:

```
backup_dev_type = disk_copy|stage_copy
```

```
stage_copy_cmd = rcp|ftp
```

For more information, see [backup_dev_type \[Page 447\]](#) or [stage_copy_cmd \[Page 470\]](#).



23.2.5 Standby Database: Remote Database Connect Requirements

23.3 Split-Mirror Disk Backup

Use

Certain conditions are required for the connection to a remote host with the [Oracle Standby Database \[Page 162\]](#). You must be able to manage the primary Oracle instance from the standby instance, that is, you must be able to start up and shut down this instance from the standby instance. You must be able to perform these operations from a local `SQLPLUS` session.

Prerequisites

[BRARCHIVE \[Page 383\]](#) and [BRBACKUP \[Page 360\]](#), which are started on the standby database server, connect remotely to the primary database. Therefore, make sure that the instance string in the [primary_db \[Page 463\]](#) parameter is defined to Oracle SQL*Net in the `tsnnames.ora` file.

Test the connection in `SQLPLUS` with the following command:

```
connect system/<password>@<value_of_primary_db>
```

Procedure

1. Create an Oracle password file:

```
orapwd file=<ORACLE_HOME>/dbs/orapw<DBSID>  
password=<sys_password> entries=100
```

2. Set the `remote_login_passwordfile` parameter in the `init<DBSID>.ora` as follows:

```
remote_login_passwordfile = exclusive
```

If the parameter is entered only after an instance has been started up, you must restart it, so that the parameter becomes effective.

3. Start as user `SYS` and execute the Oracle command:

```
grant sysoper to system;
```

This grants the `system` user `SYSOPER` authorization on the primary database instance.

4. If necessary, change the password for the `system` user:

```
alter user system identified by <password>;
```



Starting with SAP Release 4.5B, it is no longer necessary to give the `SYSTEM` user `SYSOPER` authority in the Oracle password file of the production database.

23.3 Split-Mirror Disk Backup

Use

This section describes how you can use split-mirror disk backup with [BRBACKUP \[Page 360\]](#) to perform an online or offline backup of your Oracle database without downtime. We recommend this especially for online backup of large databases.



We provide this information on split-mirror disk backup for advice only. Consult your hardware partner for help in setting up split-mirror disk backup. It is especially important to make sure that the data on the backup database host is consistent when you perform the backup.

Integration

- Computing Center Management System (CCMS)
 - You **cannot** use CCMS to **schedule** split-mirror disk backups. This is because BRBACKUP runs on the backup database host.
 - However, you can use CCMS to **monitor** backups on the production database, because BRBACKUP sends status information to the production database, which CCMS can then use.
- The BACKINT interface is also supported in the split-mirror configuration. This configuration is transparent to BACKINT and there are no extra considerations for BRBACKUP.
- A split-mirror disk backup only includes database backups with BRBACKUP. Backups of the offline redo log files using [BRARCHIVE \[Page 383\]](#) are **not** included because they do not place any significant load on the production host. In a standard setup, you back up the offline redo log files on the database server to local or remote tape devices, on disk or with BACKINT.

If the backup device is connected to the backup server, you can use the remote device – using [backup_dev_type \[Page 447\]](#) = pipe|pipe_auto|pipe_box – or the BACKINT interface.
- To synchronize BRBACKUP sessions with BRARCHIVE sessions, you can start a BRARCHIVE session immediately after the disks are split and not just at the end of the online BRBACKUP session. When the disks have been split, Oracle regards the backup as already finished.

Features

In the split-mirror configuration, the disks of the production database host are mirrored, that is, synchronized. BRBACKUP runs on a backup database host, where the backup is performed after the mirror disks have been split and mounted. On the production host, this saves processing power otherwise required for the backup, so that the production SAP System is unaffected by the backup.

BRBACKUP can be used to control the splitting and later synchronization of the disks. If you want to open the database on the backup host and use it as a "reporting server" you can perform the synchronization yourself later.

The actual splitting and later synchronization of disks is executed by a script or program supplied and supported **not** by SAP, but by the manufacturer of the operating system, disk subsystem, or backup software.

If the split disks do not have to be resynchronized immediately after the backup you can use the backup server along with the mirror disks to operate an independent SAP System on the backup server. For this you must install the entire Oracle server software on the backup server.

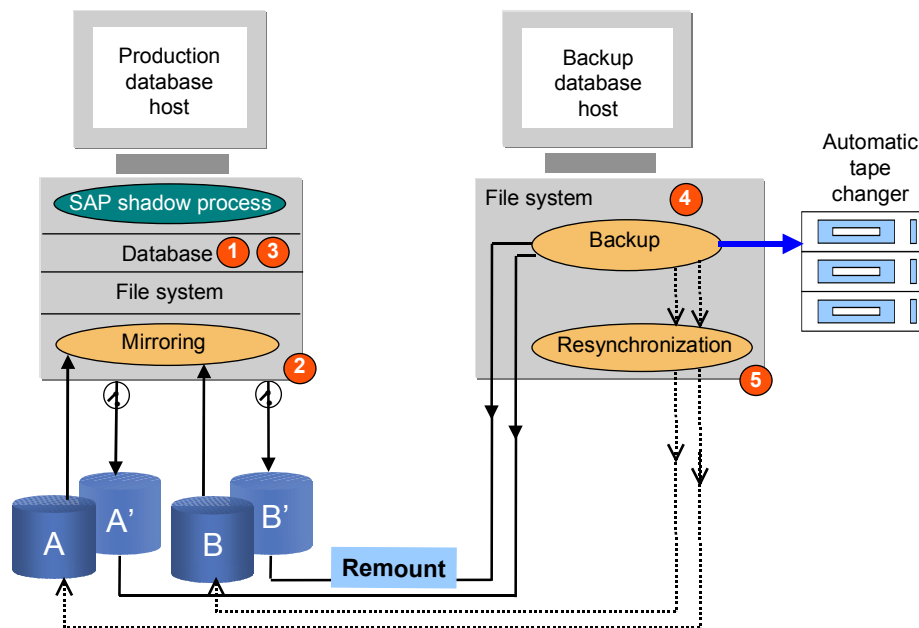
For more information, see the following:

- To start split-mirror backup on the production database server, see SAP Note 170607.
- To divide the split mirrors, see SAP Note 378818.

Activities

The following graphic shows how split-mirror backup works:

23.3 Split-Mirror Disk Backup



As shown in the above graphic, a split-mirror backup works as follows:

	Split-Mirror Online Backup [Page 172]	Split-Mirror Offline Backup [Page 173]
1	Tablespaces set to status BACKUP	Production database shut down
2	Mirror disks (A' and B' in the graphic) split and connected to backup server using <code>split_cmd</code>	
3	Tablespaces reset to normal status	Production database restarted
4	Mirror disks backed up on backup host	
5	Primary and mirror disks resynchronized using <code>resync_cmd</code>	



Consider online split-mirror disk backup if high availability is an important consideration for your system.

23.3.1 Split-Mirror Online Backup

Use

This section describes how you can use [BRBACKUP \[Page 360\]](#) to perform a [split-mirror \[Page 170\]](#) online backup with your Oracle database. We especially recommend this for large databases where high availability is an important consideration.

In an online backup the system remains available the whole time. The tablespaces to be backed up are temporarily set to BACKUP status 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 amount of redo information is significantly reduced.

Prerequisites

You set the following parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

- [backup_type \[Page 450\]](#) = `online_split`

Instead of the `backup_type` parameter, you can enter the BRBACKUP command option `-t | -type` as shown below in "Activities." The BRBACKUP command option takes priority.

- [split_cmd \[Page 470\]](#)
- [resync_cmd \[Page 470\]](#)

Activities

You start BRBACKUP on the backup host using the following command :

```
brbackup -t online_split
```

For more information, see [-t|-type \[Page 374\]](#).

BRBACKUP then performs the following steps:

1. Sets the tablespaces to be backed up to BACKUP status using
`ALTER TABLESPACE <tablespace name> BEGIN BACKUP.`
2. Splits the mirror disks using [split_cmd \[Page 470\]](#).
3. Resets the tablespaces to the normal status using
`ALTER TABLESPACE <tablespace name> END BACKUP.`
4. Backs up the mirror disks.
5. If you have set [resync_cmd \[Page 470\]](#), synchronizes the mirror disks with the originals.

See also:

[Split-Mirror Backup: Software Configuration \[Page 174\]](#)

23.3.2 Split-Mirror Offline Backup

Use

This section describes how you can use [BRBACKUP \[Page 360\]](#) to perform a [split-mirror \[Page 170\]](#) offline backup with your Oracle database.

In an offline backup the database is shut down to split the mirror disks. Therefore, it is considerably faster than a normal offline backup, in which the database is shut down for the duration of the entire backup.

Prerequisites

You set the following parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

- [backup_type \[Page 450\]](#) = `offline_split`

Instead of `backup_type`, you can enter the BRBACKUP command option `-t | -type` as shown below in "Activities." The BRBACKUP command option takes priority.

- [split_cmd \[Page 470\]](#)
- [resync_cmd \[Page 470\]](#)

Activities

You start BRBACKUP on the backup host using the following command :

```
brbackup -t offline_split
```

For more information, see [-t|-type \[Page 374\]](#).

23.3 Split-Mirror Disk Backup

BRBACKUP then performs the following steps:

6. Shuts down the database on the productive host, in order to guarantee the consistency of the database for an offline backup.
7. Splits the mirror disks using [split_cmd \[Page 470\]](#).
8. Restarts the database on the production host, so that it is again available for the SAP System.
9. Backs up the mirror disks offline on the backup host.
10. If you have set [resync_cmd \[Page 470\]](#), synchronizes the mirror disks with the originals.

See also:

[Split-Mirror Backup: Software Configuration \[Page 174\]](#)

23.3.3 Split-Mirror Backup: Software Configuration

This section describes the software configuration for split-mirror backup:

- At least the client software and `SQLPLUS` components of the Oracle database software must be installed on the backup server.
- The Oracle home directory structure must correspond to the SAP standard installation required by BRBACKUP.

Default Directories

UNIX	Windows
<code>\$ORACLE_HOME/dbs</code>	<code>%ORACLE_HOME%\DATABASE</code>
<code>\$ORACLE_HOME/bin</code>	<code>%ORACLE_HOME%\BIN</code>
<code>\$SAPDATA_HOME/sapbackup</code>	<code>%SAPBACKUP%</code>
<code>\$SAPDATA_HOME/saparch</code>	<code>%SAPARCH%</code>
<code>\$SAPDATA_HOME/sapreorg</code>	<code>%SAPREORG%</code>
<code>\$SAPDATA_HOME/sapcheck</code>	<code>%SAPCHECK%</code>
<code>\$SAPDATA_HOME/saptrace</code>	<code>%SAPTRACE%</code>



With SAP Releases older than 4.5B, the `SAPBACKUP` directory must be mounted to the backup server and the production server for the system to be able to access the control files on the backup server created by the SQL command `alter database backup controlfile to <filename>` on the production server.

Starting with SAP Release 4.5B, this is no longer necessary.
For more information, see SAP Note 156704.

- The Oracle and BRBACKUP profiles should be available in `$ORACLE_HOME/dbs` or `%ORACLE_HOME%\DATABASE`.
- Set the [backup_type \[Page 450\]](#) parameter in the `init<DBSID>.sap` profile to `offline_split` | `online_split`.

- To establish the connection between the backup server and the production server, you must define the profile parameter [primary_db \[Page 463\]](#) for the SQLNET connection. Before this, you must perform the following steps:
 - Create an Oracle password file:


```
orapwd file=<ORACLE_HOME>/dbs/orapw<DBSID>
password=<SYS password> entries=100
```
 - Set the `remote_login_passwordfile` parameter to `exclusive` in all `init<DBSID>.ora` profiles.
 - Give the system user `SYSOPER` authorization in the production database. Start `SQLPLUS` as user `SYS` and execute the following Oracle command:


```
SQLPLUS> grant sysoper to system;
```
 - If needed, change the password for the system user:


```
SQLPLUS> alter user system identified by <password>;
```
- Starting with SAP Release 4.5B, the `SYSOPER` authorization and the password file are only required to stop the primary database for the offline split. For an online split, the standard remote connection using the instance string specified in the [primary_db \[Page 463\]](#) parameter is sufficient.
- Make sure that the directory `/usr/sap/<SAPSID>/SYS/exe/run` or `\\<Host name>\sapmnt\<SAPSID>\SYS\exe\run` is accessible from the backup server and that it contains at least the programs `BRBACKUP`, `BRCONNECT`, and `BRTOOLS` (and optionally `BRARCHIVE` and `BRRESTORE`).



With SAP Releases older than 4.5B, make sure that the paths of all database files accessed by the production database and backup database are identical. Starting with SAP Release 4.5B, this restriction no longer applies. You can now mount the database files on the backup host in a different `SAPDATA_HOME` directory by using the `init<DBSID>.sap` parameter [orig_db_home \[Page 463\]](#). By using this parameter you can even mount the split-off files on the same host.

The manufacturer of the external backup tool must install and configure the `BACKINT` interface on the backup server.

Profile Parameters and Command Options for `init<DBSID>.sap`

Make sure that the following parameters are set in the [initialization profile `init<DBSID>.sap` \[Page 445\]](#):

- [backup_type \[Page 450\]](#) = `offline_split|online_split`
- [split_cmd \[Page 470\]](#) = "`<split_cmd> [$]`"

`<split_cmd>` is a program or shell script called by `BRBACKUP` to split the mirror disks.
- [resync_cmd \[Page 465\]](#) = "`<resync_cmd> [$]`"

`<resync_cmd>` is a program or shell script called by `BRBACKUP` to resynchronize the mirror disks. If this parameter is not set, `BRBACKUP` does not perform the resynchronization. At runtime, `BRBACKUP` replaces the optional character `$` with the name of the text file that contains the names of all files to be split or resynchronized.

23.4 Backup with Automatic Tape Changers



If `split_cmd/resync_cmd` is completed successfully, an exit code of 0 is returned. Only messages beginning with `#INFO` are accepted, that is, these are the only messages not interpreted as error messages. If the command is not successful, a return code of `> 0` is returned as well as messages describing the cause of the error.

- [primary_db \[Page 463\]](#) = `<inst_str>`
`<inst_str>` is an instance string to the production database to connect from the backup server to the production server. The connections are defined in the Oracle configuration file `tnsnames.ora`.
- `brbackup -t|type online_split` Online backup of mirror disks
`brbackup -t|type offline_split` Offline backup of mirror disks
 For more information, see [-t|type \[Page 374\]](#). This command line option overrides the backup type set in the `backup_type` parameter.

23.4 Backup with Automatic Tape Changers

Use

In addition to standard backups to local disks or tapes, [BRBACKUP \[Page 360\]](#) supports several types of Oracle backup with automatic tape changers, such as autochangers, autoloaders, jukeboxes, or tape robots.

Features

The following table shows the types of backup supported by BRBACKUP:

Backup type		Parameter
Local disks		disk
Local tape devices with manual tape swapping		tape
Remote tape devices with manual tape swapping		pipe
Automatic sequential tape swapping: autochangers or autoloaders	Local tape devices	tape_auto
	Remote tape devices	pipe_auto
Fully automatic tape swapping: jukeboxes or tape robots	Local tape devices	tape_box
	Remote tape devices	pipe_box

The following commands are used to define automatic mounting and dismounting of tapes in the backup device:

- [mount_cmd \[Page 459\]](#)
- [dismount_cmd \[Page 458\]](#)

These commands enable you to administer hundreds of tapes in jukeboxes and autoloaders. For more information, see [Mount and Dismount Commands \[Page 177\]](#).

Activities

You can set the parameters from the above table in either of the following ways:

- In the [backup_dev_type \[Page 447\]](#) parameter of the [Initialization Profile init<DBSID>.sap \[Page 445\]](#)

23.4 Backup with Automatic Tape Changers

- In the `-d|-device` [Page 367] command option of BRBACKUP (this overrides the initialization profile)

Some devices with automatic tape changing require additional time to change the tape. This additional time can be taken into account by adjusting `rewind_offline` [Page 466]. To do this, you can enter a command like the following:

```
rewind_offline = "mt -t $ offline && sleep 60"
```

See also:

[Autoloader Backup Example \[Page 178\]](#)

23.4.1 Mount and Dismount Commands

The mount and dismount commands are relevant if you use [jukeboxes or tape robots](#) [Page 176] for your Oracle database.

BRBACKUP [Page 360] or BRARCHIVE [Page 383] does the following:

1. Before accessing tapes for the first time, they perform the command defined in [mount_cmd](#) [Page 459] to automatically mount the tapes.
2. When the backup has finished, BRBACKUP or BRARCHIVE switches the corresponding tape devices to offline mode.
3. Then they call the command defined in [dismount_cmd](#) [Page 458] to automatically dismount the tapes.



For some tape devices, the tapes might not be switched to offline mode. The dismount command is then sufficient to dismount the tapes. In this case, do **not** set [rewind_offline](#) [Page 466].

The mount and dismount commands used in most implementations require special control drivers, which are defined in the parameters [tape_address_ctl](#) [Page 476] or [tape_address_ctl_arch](#) [Page 476].

Syntax of the commands mount and dismount commands:

```
mount_cmd = "<mount_cmd> <A> <B> <C> [<D>]"
```

```
dismount_cmd = "<dismount_cmd> <A> <B> [<D>]"
```

The command options <A> to <D> mean the following:

- <A> 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 -b | -b1 | -b2
```

where <DBSID> is the Oracle system ID (that is, the database instance)

This option can be used to identify a subset of tapes ("sub-pool") for database backup. The additional extension -A and -B enables you to define and administer separate tape sub-pools for BRARCHIVE and BRBACKUP.

- identifies the tape devices, on which the mount or dismount operations are to be performed, using the control driver addresses:

```
<dev_addr1>[, <dev_addr2>, ...]
```

where <dev_addr> is the driver address of the tape unit

23.4 Backup with Automatic Tape Changers

One or more of the tape devices defined in the parameter [tape_address_ctl \[Page 476\]](#) can be addressed.

- <C> defines the tape names to be mounted on the tape devices given in :

```
<tape name1>[, <tape name2>, ...] | SCRATCH
```

where <tape name> is the name of the tape to be mounted

The tape names are chosen from the backup volume list [volume_backup \[Page 481\]](#) (from BRBACKUP) or [volume_archive \[Page 480\]](#) (from BRARCHIVE) by the automatic tape administration. The assignment of the tapes on the tape devices can be defined freely by the mount command. The number of tape names can be greater than the number of driver addresses. In this case any tapes from this list can be mounted on the given tape devices.

The reserved tape name SCRATCH means that any unlocked tapes (that is, those for which the expiration period has finished) can be mounted. Which of the available tapes is mounted is decided by the mount command.

- <D> 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 [mount_par_file \[Page 460\]](#) parameter or by the command option [-r|-parfile \[Page 373\]](#).



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 (except for lines starting with #INFO).

23.4.2 Autoloader Backup Example

If you are using [Backup with Automatic Tape Changers \[Page 176\]](#), this example shows how to integrate an autoloader, HP 48AL, into the backup strategy for [BRBACKUP \[Page 360\]](#) or [BRARCHIVE \[Page 383\]](#).



The configuration of the mount and dismount commands is your complete responsibility. Consult your hardware partner if in doubt.

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 tape 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. Automatic tape administration is switched on.

An extract from the profile `initC11.sap` might 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 > $SAPDATA_HOME/sapbackup/.slot
```

C shell script `dismount_tape.csh`:

```
#!/bin/csh -f
set_slot=`cat $SAPDATA_HOME/sapbackup/.slot`
mtx -d $2 -u $slot
echo $slot > $SAPDATA_HOME/sapbackup/.slot
```

These scripts must be executable and be located 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 (for example, C11B11) from the backup volume 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:

```
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 performed 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 returns it to slot 11.

The BRARCHIVE backup procedure is similar 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 or programs, such as to perform parallel backups on several tapes, or to backup several databases and form sub-pools for these in a tape robot.

23.5 Veritas Quick I/O Feature

Use

[BRBACKUP \[Page 360\]](#) and [BRRESTORE \[Page 395\]](#) now support the Veritas Quick I/O feature.

23.6 External Backup Programs



Up to now BRBACKUP and BRRESTORE considered the Veritas Quick I/O files to be raw disks. They were saved with `dd` command (if no BACKINT was used) with block size equal to Oracle block size. This could cause performance disadvantages. In addition, the files had to be created manually before BRRESTORE was started.

Features

- The Veritas Quick I/O files are automatically recognized by BRBACKUP and marked in the detail log (see Q/I/O flag in messages BR118I, BR119I and BR120I). This is the basis for processing these files through BRRESTORE.
- In native BRBACKUP backups (that is, without BACKINT), the files are copied to local and remote tape units using `dd`, to local disks using `cp` or `dd` or using `rcp` or `sapftp` onto remote disks, using [backup_dev_type \[Page 447\]](#) = `stage`. Unlike the raw disks, the block size for the `dd` command can be set to any value using parameters [dd_flags \[Page 457\]](#) and [dd_in_flags \[Page 457\]](#).
- In BRBACKUP with BACKINT, the "cdev" names are transferred to BACKINT with complete path, as in the following example.



```
/oracle/DCA/sapdata13/ddici_1/.ddici.data1::cdev:vxfs:
```

- Quick I/O files can be created with relative links, as in the following example:



```
/oracle/DCA/sapdata13/ddici_1/ddici.data1 ->
.ddici.data1::cdev:vxfs:
```

We recommend you to usually adhere to the SAP standard naming convention for normal files, as in the examples above.

- BRRESTORE creates the Quick I/O files automatically before data is actually restored. Therefore, you can use any BACKINT program (not only NetBackup by Veritas) for the database backup.

23.6 External Backup Programs

Use

The SAP tools [BRBACKUP \[Page 360\]](#), [BRARCHIVE \[Page 383\]](#), and [BRRESTORE \[Page 395\]](#) provide an interface called BACKINT that can be used to access external backup programs. You can only use this interface if the BACKINT interface program is supported by the supplier of the external backup program.

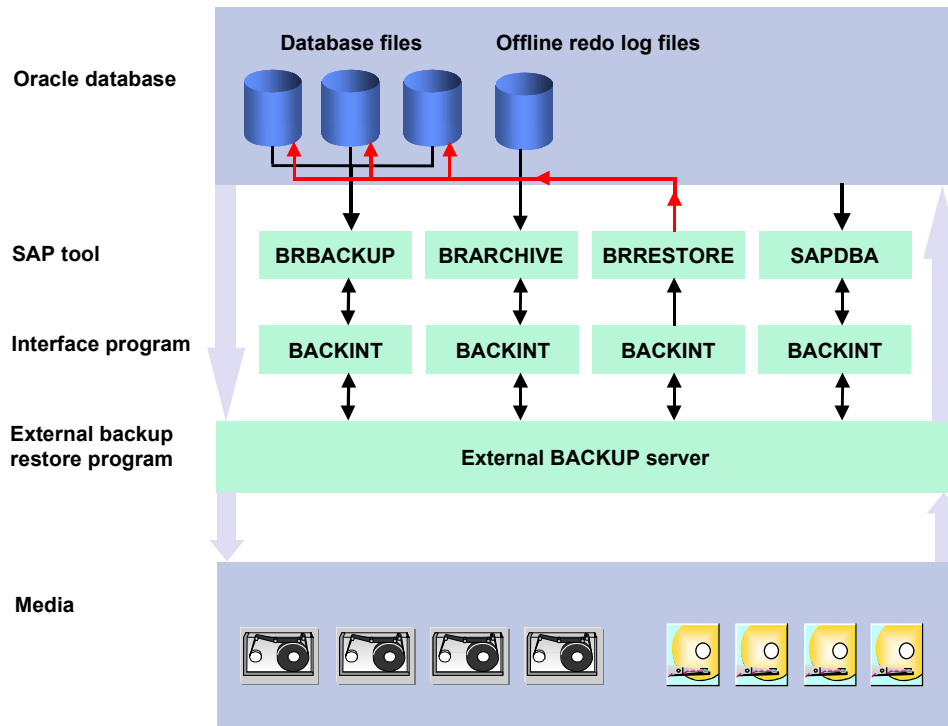
Features

By using external backup programs you gain the following advantages:

- You can use new, manufacturer-specific backup media. For example, the SAP tools do **not** support direct backup to or restore from optical storage media. However, you can use such media with an external backup program using the BACKINT interface.
- 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 a network, for example, for [backup with an automatic tape changer \[Page 176\]](#).

Activities

If you use this interface, tasks are distributed as follows:



The graphic shows that:

- BRBACKUP or BRARCHIVE does the database handling.
- The external backup program manages the backup media.
- BRBACKUP or BRARCHIVE uses BACKINT to pass a backup request to the external backup program. This request contains a list of the files for backup. BRRESTORE also uses BACKINT to trigger the external program to restore the requested files.

Any parameters that are passed are contained in a parameter file that you define. The external backup program performs all the backup operations.

- BRBACKUP, BRARCHIVE, or BRRESTORE evaluates the confirmation messages of the external backup program.



Use the following settings in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

- Device type:

```
backup_dev_type = util_file|util_file_online
```

- Parameter file, for example:

```
util_par_file = initC11.utl
```

For more information, see [backup_dev_type \[Page 447\]](#) and [util_par_file \[Page 480\]](#).

For more information about SAP partners that support the interface to external backup programs, see:

23.6 External Backup Programs

<http://service.sap.com/partners>



Tools for Oracle DBA

Use

This section describes the tools supplied by SAP for database administration (DBA) with the Oracle database.

For more information on how to work out an approach to Oracle DBA, see [Approach to Oracle DBA \[Page 66\]](#).

Prerequisites

You have already [got started with Oracle and the SAP System \[Page 23\]](#).

Features

- [SAPDBA for Oracle \[Page 184\]](#)
- [BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT \[Page 356\]](#)
- [Database Recovery with SQLPLUS \[Page 487\]](#)
- [The SAP Utilities with Windows NT \[Page 508\]](#)

24 SAPDBA for Oracle

Use

SAPDBA for Oracle is an integrated database administration tool for Oracle databases running with R/3 Systems, particularly large ones. You can use it without detailed knowledge of the database and its tools. It filters information about the database, showing you only what you need, and uses complex database statements to let you confidently manipulate the data while ensuring security and integrity.

SAP recommends you to use SAPDBA rather than the Oracle tools, because it is designed to run with R/3 databases. SAPDBA runs at operating-system level, allowing you to perform database administration tasks that are not possible when the database is running.

Integration

The following SAPDBA functions are also available in the Computing Center Management System (CCMS) of the R/3 System:

- Checking and updating cost-based optimizer statistics
- Checking the DB system (that is, checking configuration and performance)
- Monitoring of space usage in tables, indexes, and tablespaces

There are some tasks that can only be performed in CCMS (for example, scheduling backups).



When the R/3 System is running, you can use CCMS to access some of the same information (but not all) found in SAPDBA. Whether to use SAPDBA or CCMS depends on the task you want to perform, and the setup of your system.

Prerequisites

For more information about setting up SAPDBA, see [SAPDBA Configuration \[Page 185\]](#).

Features

- Graphical user interface (GUI) or command-line entry
- Analysis facilities to examine the state of the database
- Action facilities to change the state of the database
- Full logging of actions
- Menu interface in English or German
- Context-sensitive supporting information
- Available with operating systems UNIX and NT

Activities

We recommend you to implement SAPDBA if you want to perform a wide range of database administration tasks, which are more difficult or unavailable with other tools:

- Starting up and shutting down the database
- Reorganizing the database

- Administering tablespaces
- Exporting and importing objects
- Changing storage parameters
- Backing up the database
- Archiving the offline redo log files
- Restoring and recovering the database

24.1 Getting Started with SAPDBA

Purpose

This section gives you an overview of database administration with SAPDBA for Oracle. The aim is to help you get started as quickly as possible by giving you concise information and pointers to further details.

Prerequisites

If you are new to Oracle database administration with the SAP System, see [Getting Started with Oracle and the SAP System \[Page 23\]](#).

Process Flow

1. Before you start SAPDBA, you [configure SAPDBA \[Page 185\]](#).
2. When you have finished the configuration, you [start SAPDBA \[Page 193\]](#).
3. To work with logs and profiles, you look at [Management of SAPDBA Logs and Profiles \[Page 197\]](#).
4. To work with scripts, you look at [SAPDBA Scripts and Files \[Page 214\]](#).

24.1.1 SAPDBA Configuration

Purpose

This section tells you how to configure SAPDBA for Oracle before using it. The default [Initialization Profile init<DBSID>.dba \[Page 198\]](#) is `<ORACLE_HOME>/dbs/init<DBSID>.dba`. Changes to profile parameters become active when you start SAPDBA.

Prerequisites

- You have set the environment variables when you configured the database:
 - [Environment Variables \(UNIX\) \[Page 51\]](#)
 - [Environment Variables \(Windows NT\) \[Page 53\]](#)
- You are familiar with the directory structure:
 - [Directory Structure \(UNIX\) \[Page 55\]](#)
 - [Directory Structure \(Windows NT\) \[Page 60\]](#)

Process Flow

1. You [configure the display length \[Page 186\]](#) for SAPDBA.
2. You [configure the UNIX Command at \[Page 186\]](#), used by SAPDBA for reorganizations.

24.1 Getting Started with SAPDBA

3. You [set the log checks option \[Page 187\]](#).
4. You familiarize yourself with [users and roles \[Page 187\]](#).
5. You familiarize yourself with [UNIX user configurations \[Page 189\]](#).
6. You familiarize yourself with [the effects of autoextend and resize on SAPDBA \[Page 192\]](#).



Configuring the SAPDBA Display Length

Use

SAPDBA uses 24 lines and 80 columns for the screen display. You can lengthen or shorten the display. For example, if you choose to use fewer than 24 lines for the screen display then not all the SAPDBA content is displayed. However, you can still display lists correctly.

Prerequisites

The environment variables `LINES` and `COLUMNS` regulate the SAPDBA screen display. For more information, see:

- [Environment Variables \(UNIX\) \[Page 51\]](#)
- [Environment Variables \(Windows NT\) \[Page 53\]](#)

Procedure

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

Use

This section tells you how to configure the UNIX command `/usr/bin/at`, which SAPDBA uses to schedule a reorganization.

Prerequisites

The `at` command has the following authorizations:

```
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`.

Procedure

1. 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.

24.1 Getting Started with SAPDBA

2. Make sure the proper entries have been made, and then test your configuration.

For example, reorganize a small table or tablespace such as `PSAPUSER1D`. See [Reorganization \[Page 259\]](#). This helps you avoid processes crashing due to incorrect configuration of the `at` command.



Setting the Log Checks Option

Use

You can make SAPDBA log its database checks and analyses.

Procedure

1. Depending on your context, choose:
 - *Tablespace administration* → *Log checks*
For more information, see [Tablespace Administration with SAPDBA \[Page 229\]](#).
 - *Reorganization* → *Check extents and fragmentation* → *Log checks*
For more information, see [SAPDBA: Reorganization \[Page 259\]](#).
2. 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 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 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 to work with SAPDBA and BRBACKUP command options, and therefore Computer Center Management System (CCMS) DBA functions.

24.1 Getting Started with SAPDBA

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
ora<sid>	dba oper	SYSDBA SYSOPER	INTERNAL (SYS)
<sid>adm	oper	SYSOPER	OPS\$<SID>ADM
		SAPDBA	

Windows NT

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



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

OPS\$ Database User

The Oracle OPS\$ mechanism moves the entire DB security mechanism to the operating system level.

The prerequisite is that a DB user OPS\$<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:

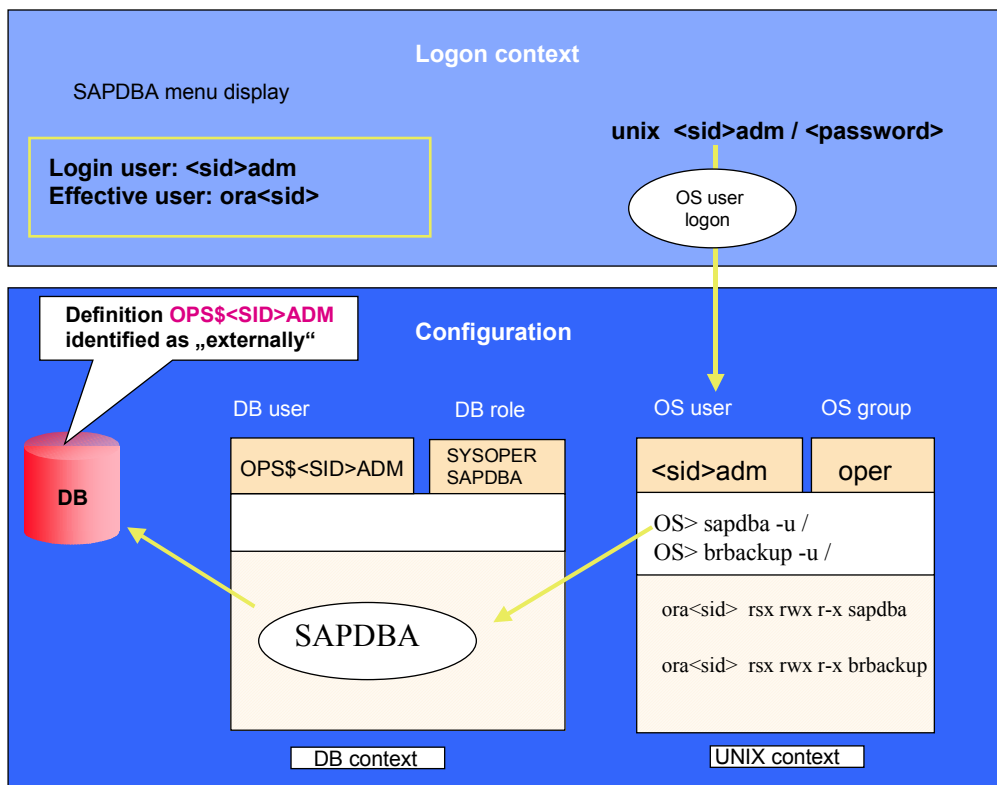
```
SQLPLUS> connect /,
```

This means you do not have to enter another password. You are then working as OPS\$<OS_user>. In the same way you can start the program SAPDBA with:

```
OS> sapdba -u /.
```

This OPS\$ mechanism is always used if you call SAPDBA or BRBACKUP from the CCMS menu in the SAP System.

24.1 Getting Started with SAPDBA



The OPS\$ 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.



UNIX User Configurations

Different user configurations at the operating system (OS) level meet different security requirements.

Configuration 1

Database administrator with all authorizations

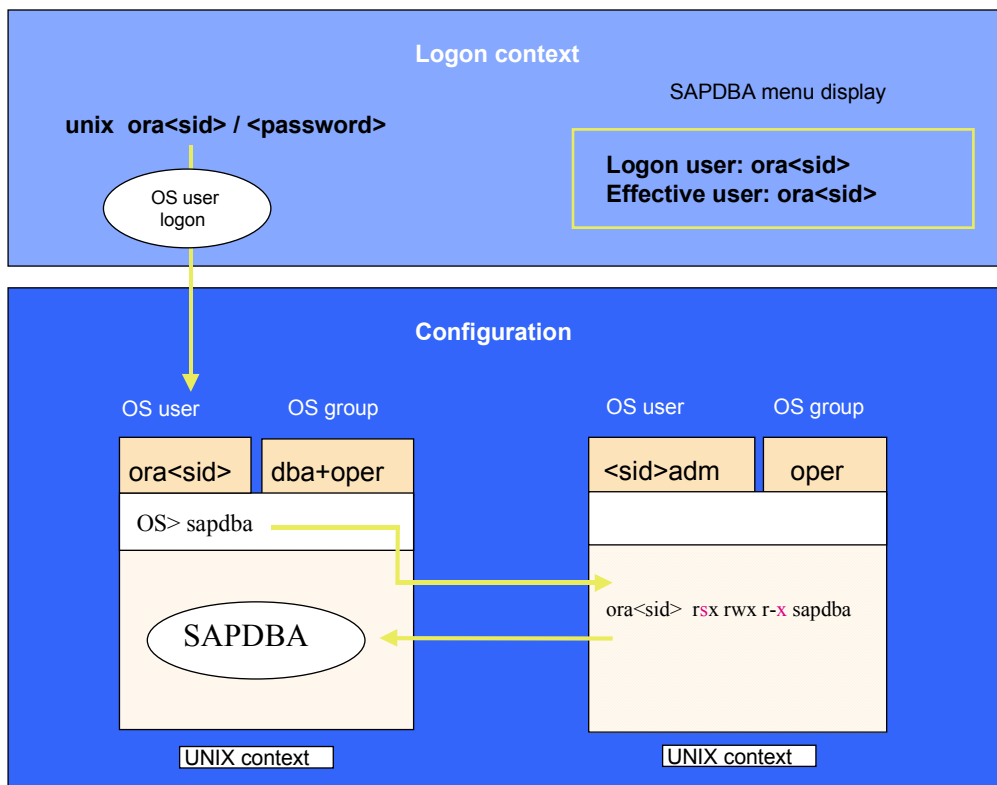
This configuration matches the SAP standard installation.

One person is fully responsible for database administration using SAPDBA and other database 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 considered and the following user configuration is appropriate:

User Configuration 1

24.1 Getting Started with SAPDBA



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.

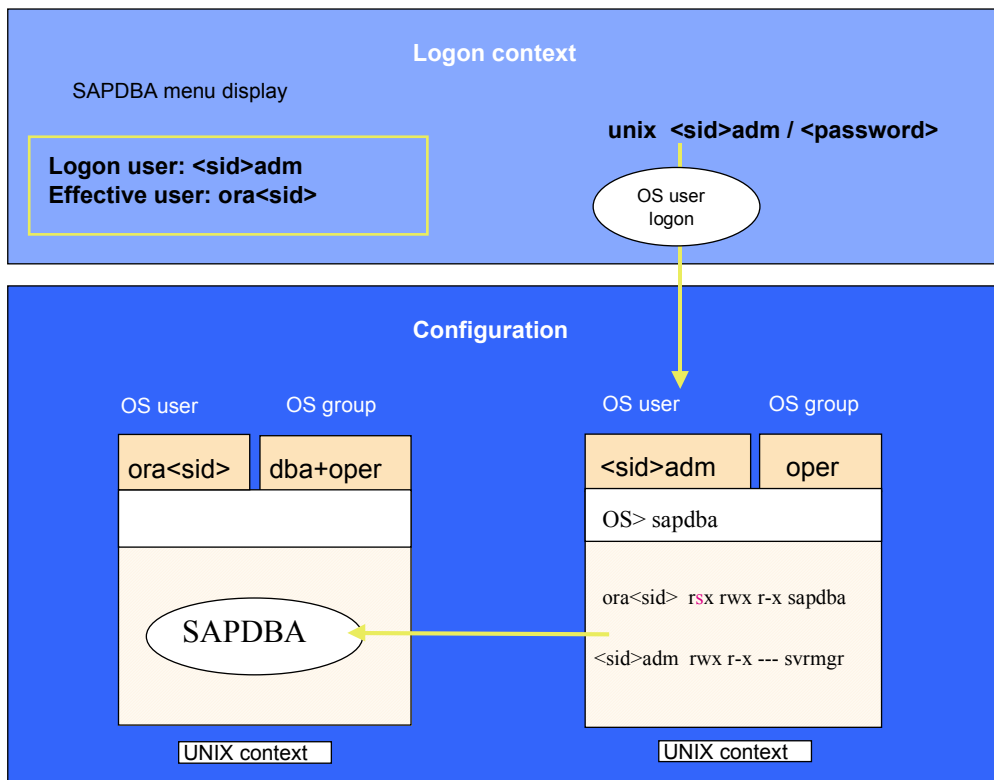
Configuration 2

User with operator privileges

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

24.1 Getting Started with SAPDBA



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 /, as in the following examples:

```
sapdba -u / -check
brbackup u / -q
```



As user <sid>adm, the operator has complete administration authorization for the SAP System, but **not** for the database. If this is not required, then you have to set up a separate OS user with the operator authorizations mentioned above. See also user configuration 3.



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

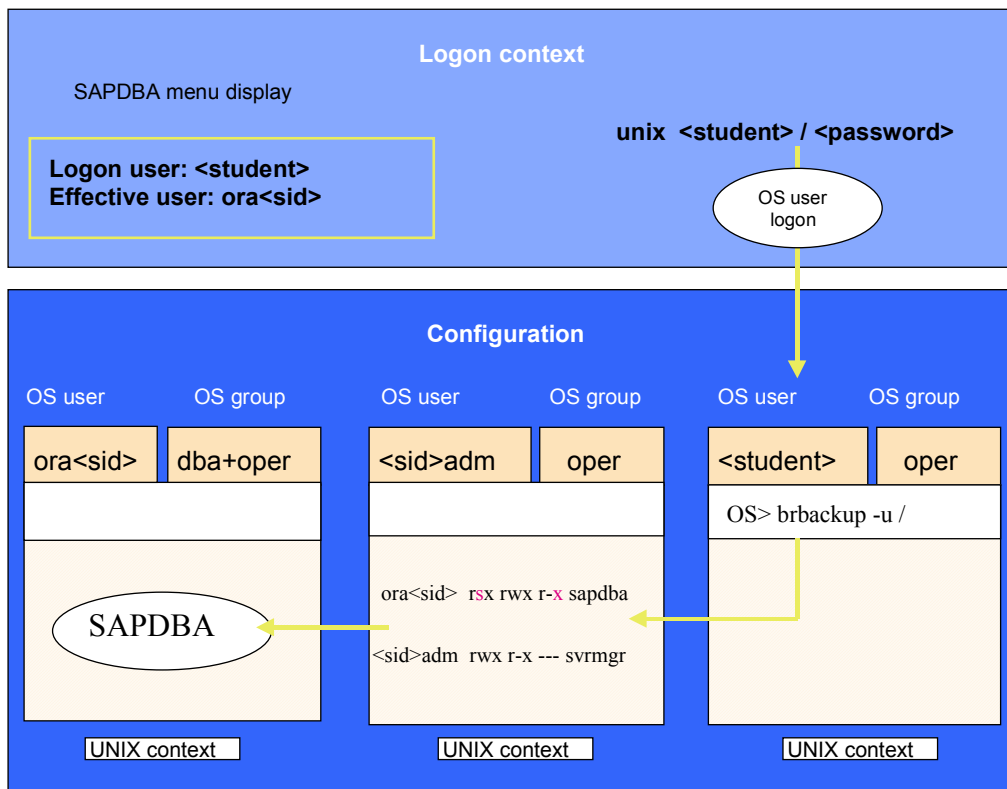
Configuration 3

24.1 Getting Started with SAPDBA

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

You need a security mechanism that allows a 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 database 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.

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 `SQLPLUS> 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.



Effects of Autoextend and Resize on SAPDBA

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

- *Autoextend* extends the data files automatically by a specified amount

24.1 Getting Started with SAPDBA

- *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 SAPDBA functions below, and have been adjusted accordingly.

Restore and Recovery

When performing a recovery (that is, 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 and after 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 are the following:

- TSP_FULL for absolute freespace
- CRITICAL_SEGS for 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 \[Page 275\]](#).

24.1.2 Starting SAPDBA

Use

You can use this procedure to start SAPDBA for Oracle. You can only use SAPDBA to manage a database system that is running on the same host system.

Prerequisites

You have configured:

- [The database system \[Page 38\]](#)
- [SAPDBA \[Page 185\]](#)

Procedure

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

24.1 Getting Started with SAPDBA



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 SAP System is `ora<sapsid>`.

You can also use SAPDBA if you are logged on as the SAP System user `<sapsid>adm`. This requires the authorization for program SAPDBA to be set accordingly (for example, under UNIX: `rwsr_xr_x oracll 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). This default user is `system`.

For more information about the user concept, see [UNIX User Configurations \[Page 189\]](#).



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

2. If you want SAPDBA to log on as user `system` to the Oracle database, call it as follows from the command line:

```
OS> sapdba
```

If the default password is not used, the system requires a password.

You can call SAPDBA with the following command 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 [SAPDBA Configuration \[Page 185\]](#). SAPDBA works with default values set internally if no profile is found.

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

Specifies the default number of extents with which SAPDBA performs checks. The extent number 2 is always proposed if no value is specified explicitly. You can find this value in the [Check Functions \[Page 232\]](#) of the menu *Tablespace administration*. These analyses are also performed if there is a check run with `sapdba -check`.

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

Specifies 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).

24.1 Getting Started with SAPDBA

Enter the password interactively. When you use `sapdba -u <user>`, the system prompts you to enter a valid password. In this case, the password does not appear on the screen as it is entered and is not 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>`

Specifies 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 unlimited.

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, 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 Main Menu \[Page 196\]](#) 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 SAP Service Marketplace)
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 \[Page 344\]](#).



Checking SAPDBA Release Information

Use

When you register problems concerning SAPDBA for Oracle with SAP support, it helps to give as much information as possible about the SAPDBA version you are using. You can display the required information using SAPDBA with the:

- Menu option *Additional functions*

24.1 Getting Started with SAPDBA

- Command option `-version`

Procedure

1. To display release information with the menu option, choose *Additional functions* → *SAPDBA Release Information*.
2. To display release information with the command option, enter the following at the command line:

```
OS> sapdba -version
```

For more information, see [sapdba -version \[Page 354\]](#).

Result

SAPDBA displays the following release information:

- *SAPDBA release*
- *SAPDBA patch level*
- *SAPDBA patch date*
- *Large files support*, that is, whether large files are supported



SAPDBA Main Menu

Definition

When you [start SAPDBA \[Page 193\]](#) for Oracle with the graphical user interface (GUI), you see the SAPDBA main menu.

Structure

The main menu looks as follows:

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		
q	Quit		

SAPDBA Commands on the Main Menu

Command	See
Startup/Shutdown instance, Instance information	Startup and Shutdown of the Oracle Database with SAPDBA [Page 217]

24.1 Getting Started with SAPDBA

<i>Tablespace administration</i>	Tablespace Administration with SAPDBA [Page 229]
<i>Reorganization</i>	Database Reorganization with SAPDBA [Page 242]
<i>Export/import</i>	Export and Import of Database Objects with SAPDBA [Page 285]
<i>Archive mode</i>	Setting Archive and Database Mode with SAPDBA [Page 41]
<i>Additional functions</i>	Starting Individual SQL Scripts with SAPDBA [Page 217] Updating Statistics using Optimizer Control with SAPDBA [Page 226] Checking SAPDBA Release Information [Page 195]
<i>Backup database</i>	Database Backup with SAPDBA [Page 303]
<i>Backup offline redo logs</i>	Backup of the Offline Redo Log Files with SAPDBA [Page 313]
<i>Restore/Recovery</i>	Performing a Full Restore and Point-in-Time Recovery with SAPDBA [Page 335]
<i>DB check/verification</i>	Partial Restore and Complete Recovery with SAPDBA [Page 323]
<i>Show/Cleanup</i>	Management of SAPDBA Logs and Profiles [Page 197]
<i>User and Security</i>	Users and Roles [Page 187] Changing Database User Passwords [Page 39] Switching to Restricted Session Mode [Page 221] UNIX User Configurations [Page 189]

24.1.3 Show/Cleanup to Manage Logs and Profiles with SAPDBA

Purpose

You manage the logs and profiles in your Oracle database with SAPDBA. You can display the logs and profiles, as well as delete them when no longer required.

Prerequisites

You need to know how to use SAPDBA and to have set it up correctly. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Process Flow

- You choose *Show/Cleanup* → *Show log files / profiles* in SAPDBA to:
 - Show log files
 - [SAPDBA log files \[Page 205\]](#)
 - Logs generated by the commands `sapdba -check`, `-checkopt`, `-analyze` or `-next`. Refer to [SAPDBA Command Mode \[Page 344\]](#).
 - [Central Log reorg<DBSID>.log \[Page 207\]](#)

24.1 Getting Started with SAPDBA

- [Structure Log \[Page 206\]](#)
- [BRBACKUP logs \[Page 377\]](#) and [BRARCHIVE logs \[Page 392\]](#)
- Show profiles:
 - [Initialization Profile init<DBSID>.dba \[Page 198\]](#) for SAPDBA
 - [Initialization Profile init<DBSID>.sap \[Page 445\]](#) for BRBACKUP, BRARCHIVE, and BRRESTORE
 - Oracle initialization profile `init<DBSID>.ora`
- You choose *Show/Cleanup* → *Cleanup log files / directories* to [delete log, directory, audit, and trace files \[Page 212\]](#) in SAPDBA as follows:
 - [SAPDBA log files \[Page 205\]](#) and directories
 - [BRBACKUP logs \[Page 377\]](#) and [BRARCHIVE logs \[Page 392\]](#)
 - Oracle trace files, audit logs

For an overview of the logs written by the SAP utilities, see [SAPDBA log files and their contents \[Page 205\]](#) and [Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT \[Page 482\]](#).



Initialization Profile `init<DBSID>.dba` with SAPDBA

Purpose

This section describes how to configure the SAPDBA program to meet your own needs.

Prerequisites

- The default directory for the initialization profile is `<ORACLE_HOME>/dbs`, but this is operating system-specific.
- In certain situations, SAPDBA starts BRBACKUP, BRARCHIVE, or BRRESTORE, so you also need to make sure that these programs are correctly set up. See [Initialization Profile `init<DBSID>.sap` with BRBACKUP, BRARCHIVE, and BRRESTORE \[Page 445\]](#).
- If the initialization profile does not exist, SAPDBA uses its own internal settings.

Process Flow

You edit the file `<ORACLE_HOME>/dbs/init<DBSID>.dba` to change the required parameter.

For more information, see [Sample Initialization Profile \[Page 204\]](#).



`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 [External Backup Programs \[Page 180\]](#).



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

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 \[Page 199\]](#).

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 \[Page 45\]](#).



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**log and Automatic archival **disabled**.

24.1 Getting Started with SAPDBA

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 \[Page 199\]](#) and/or [check_redolog_mirror \[Page 200\]](#)).



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 \[Page 199\]](#).

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 \[Page 45\]](#).



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 directory for the export dump files with this parameter. If this default value is not overwritten by the corresponding interactive entries, SAPDBA stores the export dump files in this directory.

Default value: `<SAPDATA_HOME>/sapreorg`

Possible values: A directory with sufficient space for the data to be exported. Note that this information depends on the operating system. See also [Parallel Export and Import \[Page 289\]](#).



exireo_exp_imp_buf

This parameter is used to specify the size of the buffer for export or import (used, for example, by the reorganization operation in SAPDBA).

Default value: `3000000`

Possible values: Size of the buffer in bytes. 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 [Management of Logs and Profiles \[Page 197\]](#).



exp_imp_degree

You can define the maximum possible degree of parallelism 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 and Import \[Page 289\]](#).



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
	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:

- [Displaying File and Raw Device Information for a Tablespace \[Page 242\]](#)
- [Extending a Tablespace \[Page 233\]](#)
- [Oracle Databases on Raw Devices \[Page 33\]](#)



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 \[Page 248\]](#).



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 BRTOOLS to split the files or to increase the

24.1 Getting Started with SAPDBA

number of target directories. If BRTOOLS is used, `max_file_size` defines the maximum size of the split files.

Default value: 2G



message_buf

This parameter determines the size of the message buffer.

Default value: 100000

Possible values: Size of the buffer in bytes.



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.



parallel_instances

This parameter is only relevant if you are using Oracle Parallel Server (OPS).

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)
```



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.

24.1 Getting Started with SAPDBA



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

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 $2 \cdot m$, 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

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_tsname

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 >

24.1 Getting Started with SAPDBA

<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

This is example of an initialization file, `initC11.dba`:

initC11.dba

<code>exireo_dumpdir</code>	<code>\$SAPDATA_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>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>



SAPDBA Log Files and Their Contents

The following table shows all the SAPDBA log files and what they contain

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 → Restore one tablespace</i>) or restoring individual files (<i>Restore/Recovery → Restore individual file(s)</i>)
<timestamp>.rch	Log of the check function (in submenus of <i>Reorganization → Check extents and fragmentation</i>)
<timestamp>.ral	Log of function: configuration of storage parameters (<i>Reorganization → Alter/Show table or index storage parameters</i>)
<timestamp>.rsi	Log of function: reorganize a single table or index (<i>Reorganization → Reorganize single table or index</i>)
<timestamp>.rli	Log of function: Reorganize a list of objects (<i>Reorganization → Reorganize list of tables or indexes</i>)
<timestamp>.rtc	Log of function: Reorganization of a tablespace (<i>Reorganization → Reorganize tablespace</i>)
<timestamp>.rtd	Log of function: reorganize a tablespace with data files (<i>Reorganization → Reorganize tablespace and datafiles</i>)
<timestamp>.rmv	Log of function: change the data file assignment of a tablespace (<i>Reorganization → Move/rename data files of a tablespace</i>)
<timestamp>.rrs	Log of function: Resize data file (<i>Reorganization → 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 → Export tables and indexes including data</i>)
<timestamp>.exs	Log of function: export tables and index structures (<i>Export/import → Export structures of tables and indexes</i>)
<timestamp>.exg	Log of function: export other objects (<i>Export/import → Export constraints, synonyms, sequences, views, grants</i>)
<timestamp>.imd	Log of function: import with data (<i>Export/import → Import tables and indexes including data</i>)
<timestamp>.ims	Log of function: structure import (<i>Export/import → Import structures</i>)
<timestamp>.chk	Statistics generated by using the SAPDBA command option -check (e.g. display using <i>SAPDBA daily check log files</i>)

24.1 Getting Started with SAPDBA

<code><timestamp>.nxt</code>	Log generated by using the SAPDBA command option <code>-next</code> (e.g. display using <i>SAPDBA daily check log files</i>)
<code><timestamp>.aly</code>	Log generated by using the SAPDBA command option <code>-analyze</code> (e.g. display using <i>SAPDBA daily check log files</i>)
<code><timestamp>.opt</code>	Log generated by using the SAPDBA command option <code>-checkopt</code> (requirements planning for the DB optimizer).
<code><timestamp>.brb</code>	Log of the BRBACKUP call by SAPDBA (<i>Backup database</i>)
<code><timestamp>.bra</code>	Log of the BRARCHIVE call by SAPDBA (<i>Backup archive log</i>)
<code><timestamp>.rcv</code>	Log of function: recovery (<i>Check (and repair) Database</i>)
<code><timestamp>.rsn</code>	Log of function: recovery to current time (now) (<i>Restore/Recovery → Full restore and recovery</i>)
<code><timestamp>.rsp</code>	Log of function: point in time recovery (<i>Restore/Recovery → Full restore and recovery</i>)
<code><timestamp>.rst</code>	Log of function: restore the database (<i>Restore/Recovery → Full restore</i>)
<code>esc<TSP>.log</code> <code>esc<TSP>_<n>.log</code>	BRTOOLS logs (n=1,...,number of processes). Unlike the export, import logs, these are not appended to the main log <code><timestamp>.<ext></code> but are generated separately in the subdirectory <code><timestamp></code> of the working directory.
<code>exp imp<TSP>_<n>.log</code>	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 Logs for Parallel Export and Import Processes [Page 210] .
<code>reorg<DBSID>.log</code>	Central log with overview information on the operations performed. See Central Log reorg<DBSID>.log [Page 207] .
<code>struct<DBSID>.log</code>	Structure log containing information on the structure of the database. See Structure Log [Page 206] .



Structure Log

Definition

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.

Use

This information helps you keep an accurate overview of the structure of the database. 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 is 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.

Structure

Entry	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).



Central Log reorg<DBSID>.log

Definition

The central log, `reorg<DBSID>.log`, contains information on important database 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 helps you plan your reorganizations more efficiently.

Use

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*. For more information, see [Extent Check Functions \[Page 246\]](#). However, SAPDBA instead writes detail logs (`<timestamp>.rch`) and you can display the latest analysis by using the check functions of the SAPDBA reorganization menu (for example, *Reorganization* → *Check extents and fragmentation* → *Estimate/Compute table* → *Get last analysis*).

Structure

`reorg<DBSID>.log` contains the following information:

- M

Mode of action

Possible values: N for normal, B for batch, R for restart

The operation was performed in normal mode (N) or in the background (B) or started or restarted at a later time (R).

24.1 Getting Started with SAPDBA

- LOG_TIME
<timestamp> in YYMMDDHHMM format
Logging time
- BEGIN
<timestamp> in YYMMDDHHMM format
Time process started
- END
<timestamp> in YYMMDDHHMM format
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

- 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

- SIZE_B <n> (in KB)
Size of the object before the start of the operation. This 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. This 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. This 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.

- RC

[Return Codes in the Central Log reorg<DBSID>.log \[Page 209\]](#)

Integration

For more information, see [Logging Information in CCMS \[Page 211\]](#).



Return Codes in the Central Log reorg<DBSID>.log

Definition

This section contains the return codes for the central log `reorg<DBSID>.log`.

Use

You can use the return codes to find out how database operations completed.

Structure

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)
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)
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
-------------	---------

24.1 Getting Started with SAPDBA

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



Logs for Parallel Export and Import Processes

24.1 Getting Started with SAPDBA

Definition

This section describes the logs generated when export or import processes run in parallel. For more information, see [Parallel Export and Import \[Page 289\]](#).

Use

Export and import processes can run in parallel when the following functions are used in SAPDBA:

SAPDBA Function	Main Log
<i>Reorganization → Reorganize list of tables or indexes</i>	<timestamp>.rli
<i>Reorganization → Reorganize tablespace</i>	<timestamp>.rtc
<i>Reorganization → Reorganize tablespace and datafiles</i>	<timestamp>.rtd
<i>Export/import → Export tables and indexes including data</i>	<timestamp>.exd
<i>Export/import → Import tables and indexes including data</i>	<timestamp>.imd

Structure

When several parallel export or import processes are started, partial, temporary logs are written in addition to the main logs listed above. The temporary logs are stored in the directory:

```
<working directory>/<timestamp>
```

The temporary logs are called:

```
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 or 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.

Integration

If SAPDBA cancels an import or export in progress, the partial logs are kept and not appended to the main log.

**Logging Information in CCMS****Definition**

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 [central log reorg<DBSID>.log \[Page 207\]](#), but also in database tables SDBAH and SDBAD.

Use

You can view the details stored in SDBAH and SDBAD from the Computing Center Management System (CCMS). For more information about CCMS, see [SAP/Oracle DBA in CCMS \[Ext.\]](#).

SDBAH and SDBAD are also used by the SAP utilities BRBACKUP and BRARCHIVE. For more information, see [Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT \[Page 482\]](#).

24.1 Getting Started with SAPDBA

The results of the analyses (Compute/Estimate statistics, validate structure) are also stored in the relevant database tables, so that these statistics can also be viewed from CCMS.

It is important that only SAPDBA, BRBACKUP, BRARCHIVE, or CCMS writes to these tables.

Structure

The information from the main log, for example, is stored in the following fields of SDBAH and SDBAD:



The following information is given for completeness. CCMS presents this data in a 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



Deleting Log, Directory, Audit, and Trace Files

Use

It is important that you regularly delete old log, directory, audit 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 that can be hard to analyze. To free more space, be sure to regularly delete the operating system audit trail.

Prerequisites

Log and Trace Files

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

- Logs and any directories of SAPDBA, BRBACKUP, or BRARCHIVE

24.1 Getting Started with SAPDBA



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 (CCMS) can no longer be displayed.

Note that, if a backup for which the corresponding log has been deleted needs to be restored with SAPDBA, you must first restore the log.

Always make sure that there is a sufficiently large number of logs present. The SAPDBA recommended value is to 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 until deletion of the logs or directories, the default values from the profile `init<DBSID>.dba` are used. For more information, see [Sample Initialization Profile \[Page 204\]](#).

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. The ALERT file contains very important information, so SAPDBA does **not** provide an option for deleting this file.

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, Oracle writes an auditing log for every logon to the database. As a result, the directory can grow very quickly. Be sure to regularly delete the old audit logs.

Procedure

1. To check whether the parameters beginning `expir_period` in `init<DBSID>.dba` are configured correctly, choose *Show/Cleanup* → *Show log files / directories* → `<DBSID>.dba` (SAPDBA profile) in SAPDBA

For more information, see [Initialization Profile `init<DBSID>.dba` \[Page 198\]](#).

2. To delete files using the SAPDBA menu, choose *Show/Cleanup* → *Cleanup log files / directories*.
3. Choose the appropriate option, depending on what type of object you want to clean up.
4. To delete objects using the command line, enter the following:

```
sapdba -cleanup
```

24.1 Getting Started with SAPDBA

SAPDBA removes objects that are older than the specified interval (in days) according to the defaults set in the initialization profile `init<DBSID>.dba`.

24.1.4 SAPDBA Scripts and Files

Definition

SAPDBA generates scripts containing the instructions required to perform functions such as export, import, and reorganization. SAPDBA automatically starts each required script.

For more information on a script, refer to the section that describes the corresponding function. If an error occurs, you can use the Restart function to resume the reorganization or the export or import. If this attempt also fails, you can start the required scripts manually. See "Starting the Scripts" below.

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 `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>` 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 specified by one of the following:

- the `init<DBSID>.dba` parameter [exireo_dumpdir \[Page 200\]](#) (this has the highest priority)
- `<SAPDATA_HOME>/reorg`
- `<SAPREORG>`, if defined

If several export dump destinations were defined, there are a corresponding number of export dump files, export and import shell scripts, and parameter files. See "Information in the Export Dump Destination – Script Names" below.

Structure

Overview of SAPDBA 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. It 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

24.1 Getting Started with SAPDBA

	reorganizations.
nam<TSP>.sql	Script for temporarily renaming tables during a reorganization.
rnm<TSP>.sql	Script for temporarily renaming a table during the a reorganization, with delayed deletion of the data (deferred mode).
ren<TSP>.sql	Script containing the necessary rename commands if the data file assignment of a tablespace is changed (move or rename). It can also be used to reset temporary renaming of tables during reorganizations.
swi<TSP>.sql	Script used during a reorganization with data files that triggers certain security mechanisms (such as redo log file switch, backup of the control file).
tab<TSP>.sql	Script for creating tables.
tsp<TSP>.sql	Script used to create tablespaces during the reorganization of tablespaces with data files.
exc<TSP>.sh	Script for compressing (compress) or splitting (BRTTOOLS) export dump files.
exp<TSP>.sh	Oracle export of tables and their data
exc<TSP>.par	Parameter file for BRTTOOLS
exp<TSP>.par	Parameter file for export and import
exp<TSP>.dmp	Export file or named pipe (if you compress data or split it with BRTTOOLS) that contains tables with data.
exp<TSP>.dmp.Z	Compressed export file
exp<TSP>.dmp.<m> exp<TSP>.dmp.Z.<m>	Export dump files can be split into several smaller files <m> with BRTTOOLS.
imc<TSP>.sh	Script for decompressing or merging (BRTTOOLS) split export dump files for the Oracle import.
imp<TSP>.sh	Oracle import of tables and their data (including grants)
imx<TSP>.sh	Shellscript for test import (dump file check)
del<TSP>.sh	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, imx<TSP>.sh, inx<TSP>.sql

Integration

Information in the Export Dump Destination – Script Names

24.1 Getting Started with SAPDBA

- 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<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|<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 and Import \[Page 289\]](#) – both cases can be edited either sequentially or in parallel.

Use

Starting the Scripts

When an export, import, or reorganization process is terminated, you can use the restart functionality of 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.



Only use the options for starting the individual scripts described in this section in an emergency.

[Starting Individual SQL Scripts with SAPDBA \[Page 217\]](#)

[Starting SQL Scripts Manually with SQLPLUS \[Page 216\]](#)

[Starting Shell Scripts for Oracle Export and Import \[Page 217\]](#)



Starting SQL Scripts Manually with SQLPLUS

Use

You can use the Oracle program SQLPLUS to start the SQL scripts. However, this is not essential, because you can also start all scripts 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.

Procedure

Enter the following command, if necessary, to start a script with SQLPLUS:

24.2 Startup and Shutdown of the Oracle Database with SAPDBA

```
sqlplus <uid/>passwd>> @<script[.sql]>
```

The meaning of the parameters is:

- <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.



Starting Individual SQL Scripts with SAPDBA

In the initial menu of SAPDBA for Oracle, choose *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 and Import

Use

SAPDBA uses the Oracle export and import tools to export and import tables and their data. SAPDBA creates shell scripts that can be used to call the export and import programs.

Procedure

To manually perform an import of tables and their data, for example, enter a command like one of the following in the work directory:

```
OS> imp<TSP>.sh
```

```
OS> imp<TSP>_<n>.sh
```

24.2 Startup and Shutdown of the Oracle Database with SAPDBA

Purpose

You can use SAPDBA to start up and shut down the Oracle database.

Process Flow

1. You start the database using one of the following methods:
 - The SAPDBA GUI. See [Starting up the Oracle Database System \[Page 218\]](#).
 - The SAPDBA command-line entry `sapdba -startup` or `sapdba -startup_dba`. See [sapdba -startup\(dba\) \[Page 353\]](#).
 - The operating system command `startsap`.
2. If required, you [switch the database to restricted session mode \[Page 221\]](#). This is required for certain operations such as [reorganization \[Page 259\]](#).
3. You [check the database status \[Page 218\]](#).
4. You shut down the database using one of the following methods:
 - The SAPDBA GUI. See [Shutting down the Oracle Database System \[Page 219\]](#).

24.2 Startup and Shutdown of the Oracle Database with SAPDBA

- The SAPDBA command-line entry `sapdba -shutdown` or `sapdba -shutdown_abort`. See [sapdba -shutdown\(abort\) \[Page 353\]](#).
- The operating system command `stopsap`.

24.2.1 Checking the Status of the Oracle Database System

Use

You can use SAPDBA for Oracle to display the status of your database system when you do any of the following:

- Go to the main menu
- Choose *Instance information*
- Choose *Startup/shutdown instance*

These methods are described in more detail below.

Prerequisites

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Go to the [SAPDBA main menu \[Page 196\]](#).

SAPDBA displays the status of your database system:

- `ORACLE_SID`, `ORACLE_HOME` of the local instance are displayed
- `DATABASE`: The status of the local database instance is displayed (for example, *open*)
- `SAPR3`: This indicates whether a `saprr3` user is connected to the database (*connected*, *not connected*).

2. Choose *Instance information*.

SAPDBA displays the following status information:

Status information	Meaning
<code>SELECTED INSTANCE</code>	Name of the instance (for example, <i>C11</i>)
<code>INSTANCE STATUS</code>	Status of the instance (for example, <i>shutdown</i> , <i>mounted</i> , <i>opened</i>)
<code>INSTANCE HOME</code>	Local home directory (for example, <i>/oracle/C11</i>)
<code>CONNECTED PROGRAMS</code>	Number of processes connected with the database (for example, <i>1</i>)
<code>SAPR3</code>	Logon status of user <code>saprr3</code> (<i>connected</i> , <i>not connected</i>)

3. Choose *Startup/Shutdown instance* to start or stop the database instances.

With this option, SAPDBA displays status information for the local instance, the parallel instances, the number of processes logged, and user `saprr3`.

24.2.2 Starting Up the Oracle Database System

24.2 Startup and Shutdown of the Oracle Database with SAPDBA

Use

You can use SAPDBA for Oracle to start up the database system.

Prerequisites

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

You can perform some of the options below from the command line:

- *Startup normal* is equivalent to `sapdba -startup`.
- *Startup RESTRICT* is equivalent to `sapdba -startup_dba`.

For more information, see [sapdba -startup\(dba\) \[Page 353\]](#).

Procedure

1. Choose *Startup/Shutdown instance* → *Startup* in SAPDBA.
2. Choose the mode required to start up the database:

- *Startup normal*

Choose this option to start up the database instance for regular, productive operation. SAPDBA attempts to open the database instance.

- *Startup RESTRICT*

Choose this option to start the database instance in restricted session mode. This is normally used for exclusive access by the database administrator.

Start the database system in this mode, for example, if you want to [reorganize \[Page 259\]](#) 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 restricted session mode.

For information on changing to restricted session mode **without** stopping and restarting the database, see [Switching to Restricted Session Mode \[Page 221\]](#).

- *Startup NOMOUNT*

Choose this option to start only the instance, not the database itself.

- *Startup MOUNT*

Choose this option to mount the database.

The database system issues messages about the progress of the startup.

3. When the database has started, check the [status information \[Page 218\]](#).

Result

You have started the database. To shut it down, see [Shutting down the Oracle Database System \[Page 219\]](#).

24.2.3



Shutting Down the Oracle Database System

Use

You can use SAPDBA for Oracle to shut down the database system.

Prerequisites

24.2 Startup and Shutdown of the Oracle Database with SAPDBA

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

You can perform some of the options below from the command line:

- *Shutdown normal* is equivalent to `sapdba -shutdown`.
- *Shutdown abort* is equivalent to `sapdba -shutdown_abort`.

For more information, see [sapdba -shutdown\(_abort\) \[Page 353\] \[Page 353\]](#).

Procedure

3. Choose *Startup/Shutdown instance* → *Shutdown* in SAPDBA.

4. Check the [status information \[Page 218\]](#).

5. Choose the mode required to shut down the database:

– *Shutdown normal*

Choose this option to shut down the database system normally. To use this option, make sure that only one process is connected to the database instance, *Connected Programs: 1*. This means that SAPDBA is the only user process.

Use this option whenever possible.

– *Shutdown immediate*

When more than one process is logged on to the database instance, you can use this option to close the database instance.

SAPDBA does not shut down the database when:

- User `saprr3` is still linked to the database system, that is, the SAP System is still running.
- Database contents are currently being backed up, that is, BRBACKUP is currently running.

If the database system is currently archiving online redo log files to the archiving directory – that is, process ARCH is backing up an online redo log file – the shutdown process might take several minutes.

If you use *Shutdown immediate* to close the database instance, the links to all database users are 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 (the syntax varies according to 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, shut them down to avoid problems during reorganization and other operations (for example, using the UNIX command `kill <process number>`).

24.2 Startup and Shutdown of the Oracle Database with SAPDBA



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.

– *Shutdown immediate force*

Choose this option to forcibly disconnect `SAPR3` users connected to the database. Use this option with care.

– *Shutdown abort*

Choose *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*.



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. Therefore, the stopped database system has 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 is an immediate automatic recovery of the Oracle instance (instance recovery) at the next startup. Data changed after successful completion of a transaction is recovered automatically, but data changed by terminated transactions is lost. 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.

Result

You have shut down the database. To start it up, see [Starting Up the Oracle Database System \[Page 218\]](#).

24.2.4 Switching to Restricted Session Mode

Use

You can operate the Oracle database in restricted session mode. This means that only users with a specific access privilege – usually database administrators – can log onto the database.

You can use SAPDBA to check the current database mode and change it if necessary.

Prerequisites

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- For more information on how to start up the database in restricted mode, see [Starting up the Oracle Database System \[Page 218\]](#).

Procedure

1. Choose *User and Security* → *Restricted Mode*.
SAPDBA displays the current database mode, normally *RESTRICTED MODE: Disabled*.
2. Choose *Enable restricted mode* to change the database mode.
SAPDBA switches the database to restricted mode, first checking whether the *SAPR3* user is logged on to the database and warning you if so.



When you exit SAPDBA, it checks the database mode. If you have been working in restricted session mode, SAPDBA asks if you want to change the mode to normal, so enabling other users to log onto the database.

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

Use

You can update statistics on the Oracle database using SAPDBA.

By running update statistics regularly, you make sure that the database statistics are up-to-date, so improving database performance. The Oracle cost-based optimizer (CBO) uses the statistics to optimize access paths when retrieving data for queries. If the statistics are out-of-date, the CBO might generate inappropriate access paths (such as using the wrong index), resulting in poor performance.

From Release 4.0, the CBO is a standard part of the SAP System. If statistics are available for a table, the database system uses the cost-based optimizer. Otherwise, it uses the rule-based optimizer.



We **strongly recommend** you to use the [update statistics \[Page 420\]](#) functionality in BRCONNECT instead of the SAPDBA functionality. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Prerequisite

To use the CBO, make sure that the parameter `OPTIMIZER_MODE` in the Oracle initialization profile [init<SID>.ora \[Page 47\]](#) is set to `CHOOSE`.

Features

You can update statistics using one of the following methods:

- DBA Planning Calendar in the Computing Center Management System (CCMS)
For more information, see [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#). This method uses the integrated approach described below under "SAPDBA."

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA



We recommend you to use this approach instead of SAPDBA because you can schedule update statistics to run **automatically** at specified intervals (for example, weekly).

- SAPDBA

There are the following approaches to update statistics in SAPDBA:

- Integrated

This is the newer approach. It uses:

- Parallel processing to reduce the time taken to update statistics
- A single procedure to both check and then if necessary update statistics

For more information, see [Updating Statistics using Integrated Approach with SAPDBA \[Page 223\]](#).

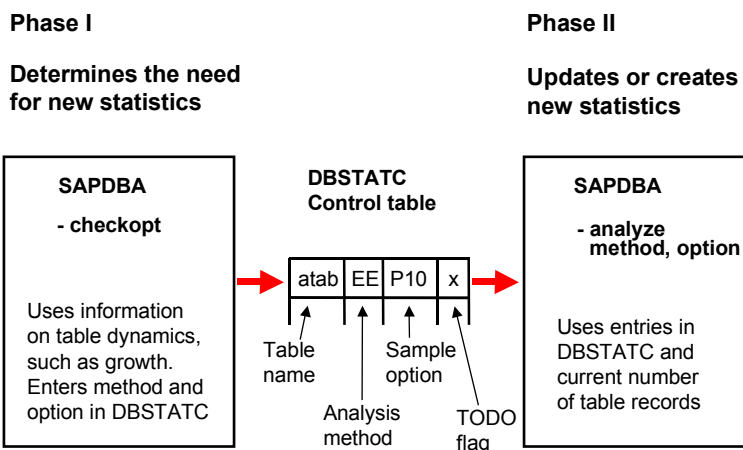


If you want to update statistics with SAPDBA, we recommend you to use the integrated approach instead of the two-phase approach.

- Two-phase

This is the older approach. It uses separate phases to check and then if necessary update statistics. Although we recommend you **not** to use this approach, it is described here in case you are still using it. Be sure to run phases I and II regularly with this approach.

The following graphic shows the two-phase approach:



For more information on how SAPDBA decides whether a table needs update statistics, see [Update Statistics: Internal Rules with SAPDBA \[Page 227\]](#).

24.3.1 Updating Statistics using Integrated Approach with SAPDBA

Use

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

You can use this procedure to check and update statistics for the Oracle cost-based optimizer (CBO), resulting in improved database performance. This section describes the new integrated SAPDBA approach for update statistics, which you start from the command line. It uses parallel processing to reduce the time taken to update statistics: the father process starts child processes to perform the optimizer analyses for individual tables, and communication between father and child uses pipes.



We **strongly recommend** you to use the [update statistics \[Page 420\]](#) functionality in BRCONNECT instead of the SAPDBA functionality. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Here is the detailed command syntax to update statistics with SAPDBA:

```
sapdba
-statistics ALL | DBSTATCO | <tables>
[ -method E | C | EH | CH
-option Pn | Rn ]
```

The command is explained in the procedure below.

For an overview of update statistics for the Oracle database, see [Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#).

Prerequisites

- Degree of parallelism

The number of parallel child processes is determined by the `cbo_parallel_degree` parameter in the SAPDBA profile `init<SID>.dba` [\[Page 198\]](#). If the parameter is not specified or zero, SAPDBA sets parallelism to the number of CPUs in the database host.

- Logging

The operation log is stored in the `SAPCHECK` directory under `<timestamp>.sta`. For more information on the `SAPCHECK` directory, refer to [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

Procedure

1. To update **all** database statistics, enter the following command:

```
sapdba -statistics
```



This is the usual command, which is equivalent to the following with the previous two-phase method:

```
sapdba -checkopt PSAP%
sapdba -analyze DBSTATCO
```

You can add the switch `ALL` to the command `sapdba -statistics`, but it is no longer necessary.

SAPDBA:

- Deletes unwanted statistics

SAPDBA deletes the statistics on tables that do not need statistics. These include all R/3 pool and cluster tables as well as all tables listed in the `DBSTATC` table for which `ACTIV` is set to `N`.

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

- Checks and, if necessary, updates statistics for normal tables (that is, tables **not** listed in the DBSTATC table)

SAPDBA proceeds as follows:

- Check statistics

SAPDBA examines the primary index of each table to see how many records have changed. The analysis method used to see how many records have changed is described in [Update Statistics: Internal Rules with SAPDBA \[Page 227\]](#).

- Update statistics

If enough records in a table have changed, SAPDBA updates statistics for the table. The criteria that SAPDBA uses to decide whether enough records have changed are described in [Update Statistics: Internal Rules with SAPDBA \[Page 227\]](#).

SAPDBA does not update statistics for the following tables:

- All tables in the DBSTATC table with ACTIV set to N or I
- All R/3 pool and cluster tables
- Generates initial statistics for new tables

SAPDBA generates statistics by analyzing the primary index using the method "Estimate Statistics Sample 1 Percent." The next time that you run update statistics with SAPDBA, it handles the table like any other table.

- Updates statistics for tables listed in the DBSTATC table

SAPDBA updates statistics for all tables in the DBSTATC table, to which the following apply:

- The TOBDO flag is set.
- ACTIV is set to U.
- ACTIV is set to A or P and enough records in the table have changed since the last update statistics.
- ACTIV is set to I, the DURAT flag is set, and enough records in the table have changed since the last update statistics.

The criteria that SAPDBA uses to decide whether enough records have changed are described in [Update Statistics: Internal Rules with SAPDBA \[Page 227\]](#).

2. To **only** update statistics for tables listed in the DBSTATC table, enter the following command:

```
sapdba -statistics DBSTATCO
```

SAPDBA follows the rules given in "Updates statistics for tables listed in the DBSTATC table" in the previous step.

3. To check and update statistics for a specific table or set of tables, enter the following command:

```
sapdba -statistics <tables>
```

You can specify <tables> using the name of a single table or a wildcard (for example, psap*).

SAPDBA proceeds as described for `sapdba -statistics` (see the first step above) for the table or tables that you specify.

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

4. To force SAPDBA to use a particular method or option for update statistics, enter a command containing `method` or `option`, such as the following:

```
sapdba -statistics ALL -method E -option P10
```

The methods and options are as follows:

- `method E` means that statistics are updated using an estimate based on a sample of records in the table.
- `method C` means that statistics are updated by computing the new statistics.
- `method EH` or `method CH` means that histograms are generated using method E or C.
- `option Pn` means that the sample is n percent (in the example above, 10 percent)
- `option Rn` means that the sample is n thousand rows

Result

The database statistics are up-to-date, resulting in better performance.

24.3.2 Updating Statistics using Optimizer Control with SAPDBA

Use

With this procedure you can display statistical information, check the requirements for new statistics, delete superfluous statistics, and create missing statistics for the Oracle database with SAPDBA. Using this procedure, you can find information on the cost-based optimizer (CBO) and control how SAPDBA updates statistics.

For more information on update statistics with SAPDBA, see [Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#).



We **strongly recommend** you to use the [update statistics \[Page 420\]](#) functionality in BRCONNECT instead of the SAPDBA functionality. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Procedure

1. To **display existing statistics**, choose *Additional functions* → *DB Optimizer control* → *Show all statistics information* in SAPDBA.

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

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

2. To **check which tables need new statistics**, choose *Additional functions* → *DB Optimizer control* → *Check for new statistics* in SAPDBA. Then choose the option you require:

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

Option	Description
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 DBSTATC
g	Start check for table
h	Start check for tables in tablespace

With these options you can:

- Specify the threshold for the percentage change in the number of records for a table using option a, *Delta (percent) for new stat. indicator*. If the threshold is exceeded, SAPDBA sets the `TODO` flag in the `DBSTATC` table so that the statistics are refreshed during the next update statistics
 - Update the entries in the control table, independently of the current status using option b, *Forced check*. This resets all existing `TODO` flags.
 - Restrict the check to a certain period of time using option c, *Time limitation*. However, do not normally set a time limit.
 - Check statistics for the tables specified in the `DBSTATC` control table, for a single table, or for all the tables in a tablespace, using options f, g, and h respectively.
3. To **delete unnecessary statistics**, choose *Additional functions* → *DB Optimizer control* → *Delete statistics excluded for R/3*.
- SAPDBA deletes tables not required for the R/3 System, such as pool and cluster tables or tables marked as not required in the `DBSTATC` control table (`Activ = N` or `R`).
4. To **create missing statistics for new tables or indexes**, choose *Additional functions* → *DB Optimizer control* → *Create missing statistics for all tables and indexes*.
- When, for example, you add one or more new tables or indexes, you can easily create statistics for them using this menu option.

24.3.3 Update Statistics: Internal Rules with SAPDBA

This section describes the rules used by the cost-based optimizer (CBO) to update statistics for the Oracle database. The rules apply to both the older two-phase approach and the newer [integrated approach \[Page 223\]](#) to update statistics with SAPDBA.



We **strongly recommend** you to use the [update statistics \[Page 420\]](#) functionality in BRCONNECT instead of the SAPDBA functionality. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Phase I: `sapdba -checkopt`

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

24.3 Update Statistics for the Cost-Based Optimizer with SAPDBA

1. SAPDBA determines the number of old table records from one of the following:
 - The Oracle Data Dictionary (table `DBA_TABLES` and `DBA_INDEXES`)
 - The `DBSTATTORA` table if there are no entries in the Oracle Data Dictionary
2. SAPDBA decides on the analysis method and sample option to use for `analyze index` in the next step as follows:

Number of old table records	Analysis method	Sample option
Records < 6,000	C	
6,000 < Records < 60,000	E	P30
60,000 < Records < 600,000	E	P10
600,000 < Records < 6,000,000	E	P3
Records > 6,000,000	E	P1

Analysis method C means compute the statistics exactly. Analysis method E means estimate the statistics using the sample option specified.

For example, “E P30” means that SAPDBA takes an estimated sample using 30% of rows.

3. SAPDBA determines the number of current table records with the command `analyze index estimate/ compute statistics` using the analysis method and sample option from the previous step.
4. SAPDBA updates the `DBSTATC` control table. It sets the analysis method and sample option based on the results of the second step. It sets the `TODO` flag if one of the following is true:
 - The current number of table records is less than 200,000 and the number has changed by at least 10%.
 - The current number of table records is greater than 200,000, less than the number of old table records, and the number has decreased by at least 10%.
 - The current number of table records is greater than 200,000, greater than the number of old table records, and the number has increased by at least 100%.



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

SAPDBA enters tables in the `DBSTATC` control table if either of the following is true:

- Statistics already exist and the change is greater than the level required to set the `TODO` flag defined in this step.
- The analysis method from the second step has changed.

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

SAPDBA deletes tables from the `DBSTATC` control table if either of the following is true:

- The date in the field `TODO chg.date` is older than 30 days, the type of usage is set to 0, the customer flag is not set, and the control flag is set to A.
- The table no longer exists in the database.

Phase II: `sapdba -analyze`

24.4 Tablespace Administration with SAPDBA

SAPDBA uses the analysis method and option specified in the `DBSTATC` control table 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 the second step 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 `ACTIV` flag set to `R` or `N`
- 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 `ACTIV` flag set to `R` are analyzed; the statistics are then deleted. The statistics results are stored in the tables `DBSTATTORA` or `DBSTATIORA`.

24.4 Tablespace Administration with SAPDBA

Use

You can use SAPDBA for Oracle to administer tablespaces.

Prerequisites

You know how to use SAPDBA and have set it up correctly. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Features

- Check for free space and fragmentation in a single tablespace or all tablespaces
- Display the assignments of tablespaces to datafiles or raw devices
- Display raw device drivers
- Add datafiles to extend existing tablespaces
- Create new tablespaces or drop empty tablespaces

Activities

1. You choose *Tablespace administration* from the SAPDBA main menu.
2. You [check tablespaces \[Page 230\]](#) regularly, distinguishing between the following main problems:
 - The tablespace is highly fragmented, requiring a [reorganization \[Page 242\]](#)
 - The tablespace has too little free storage space, requiring an extension (see below)



SAPDBA records the performed checks in the `<timestamp>.ext` log, when you have set the *Log checks* option to yes. See [Setting the Log Checks Option \[Page 187\]](#).

24.4 Tablespace Administration with SAPDBA

SAPDBA always logs the extension of a tablespace or the creation of a new one.

3. If necessary, you [extend an existing tablespace \[Page 233\]](#).
4. If necessary, you [create a new tablespace or drop an existing one \[Page 235\]](#).
5. If you have created or extended a tablespace, you [back up the database \[Page 241\]](#).
6. If necessary, you can [display file and raw device information \[Page 242\]](#) for a tablespace.

24.4.1 Check Tablespaces

Purpose

It is important to regularly monitor the tablespaces in your Oracle database, because you might need to extend or reorganize them. In either case, database performance can worsen.

The database monitor of the Computing Center Management System (CCMS) also helps you check tablespaces. Refer to [Monitoring the Oracle Database \[Ext.\]](#).

Prerequisites

These are the two main cases that you need to distinguish when checking tablespaces:

- The tablespace has too little free storage space.
You must always extend a full tablespace, otherwise it overflows if more data is added. This process describes how to check tablespaces for overflow.
- The tablespace is highly fragmented.
If the tablespace has sufficient free storage space, it is probably a fragmentation of free storage space that is causing the storage problems. This occurs only if DROP operations are performed frequently. Database objects that are frequently deleted (that is, dropped) and created again should be separated.

You can solve the storage problems by reorganizing the fragmented tablespace or the fragmented tables and indexes that it contains. An extension of the tablespace is then unnecessary.

However, a reorganization is time-consuming and complicated. Therefore, SAP recommends that you extend the tablespace rather than reorganizing it. Reorganize a 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

For more information on when database limitations require you to reorganize the database, see [Limitations of the Database System \[Page 36\]](#).



If extending a tablespace causes your database to approach the limit determined by the operating system for the number of files, we recommend you to extend then reorganize the tablespace or to extend the tablespace during the reorganization. However, a reorganization is probably unnecessary because the maximum possible number of data files is very high (at least 254 files are available).

Process Flow

24.4 Tablespace Administration with SAPDBA

1. You check the [freespace for tablespaces \[Page 231\]](#).
2. If necessary, you [check the freespace for objects in tablespaces \[Page 232\]](#).
3. If necessary, you decide how to [prevent tablespace overflow \[Page 233\]](#).

Result

By checking the tablespaces regularly, you maintain a high level of database performance.

If your checks indicate that it is necessary, you now [extend the tablespace \[Page 233\]](#).



Checking Freespace for Tablespaces

Use

You can use SAPDBA for Oracle to check the freespace for tablespaces. The information you see helps you to decide whether to extend or reorganize a tablespace.

Prerequisites

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Choose *Tablespace Administration* → *Free space and fragmentation of all tablespaces*.

SAPDBA displays details of the space situation for all tablespaces in the database. The values `Total`, `Allocated`, and `Largest` are in KB.

2. Check the following information in particular:

- `Allocated` and `%-Alloc`.

These columns refer to the reserved storage space in kilobytes and as a percentage of the tablespace size.

When `%-Alloc` is more than 90 percent, the tablespace might soon overflow. Therefore, you need to extend it by adding a new data file as soon as possible. SAPDBA lists such critical tablespaces at the end of the report.



Do **not** try to solve this problem by reorganizing the database. A reorganization only defragments the complete storage space. This means that the fragmented used storage space and fragmented free storage space are put together. That is, the storage space is reorganized, but **no additional space is gained**. In fact, additional space might even be required. Always extend a tablespace when there is little space left.

- `Free areas` and `Largest`

These columns refer to the number of freespace areas and the size of the largest freespace area.

If the percentage of the storage space used is relatively low, 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 fragmentation. In this case, it might be useful to reorganize the fragmented tablespace (or particular highly fragmented tables and indexes in it) instead of extending it.

24.4 Tablespace Administration with SAPDBA



However, note that you can usually solve this problem by adding a new file, since there are normally enough data files still available before the [limit \[Page 36\]](#) is reached. If possible, avoid reorganization because it is more complicated and time-consuming.

Result

If you have identified a tablespace that needs further investigation, refer to [Checking Freespace for Objects in Tablespace \[Page 232\]](#).



Checking Freespace for Objects in Tablespace

Use

You can use SAPDBA for Oracle to check the freespace for objects in a tablespace. The information you see helps to decide whether to extend a tablespace. It can also be useful when you [reorganize \[Page 242\]](#) the database.

You can also use this procedure to check which tablespaces would overflow if an additional number of extents were assigned.

Prerequisites

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

3. To check the objects in **all** tablespaces of the database:
 - a. Choose *Tablespace Administration* → *Check free space of objects in all tablespaces*.
SAPDBA prompts you for the threshold number of extents that would cause tablespace overflow. For example, if you specify two, then SAPDBA displays all objects that would cause overflow if two new extents were added.
 - b. Enter the threshold number of extents which would cause overflow.
SAPDBA checks all tablespaces and displays a list of the objects (that is, tables or indexes) for which the additional number of extents you specified would cause the corresponding tablespace to overflow.
2. To check the objects in a **single** tablespace:
 - a. Choose *Tablespace Administration* → *Tablespace* and enter the tablespace that you want to check.
SAPDBA validates and enters the tablespace name into the appropriate menu option.
 - b. Choose *Tablespace Administration* → *Check free space of objects in tablespace <name that you just entered>*
SAPDBA prompts you for the threshold number of extents that would cause tablespace overflow. For example, if you specify two, then SAPDBA displays all objects that would cause overflow if two new extents were added.
 - c. Enter the threshold number of extents which would cause overflow.
SAPDBA checks the chosen tablespace and displays a list of the objects (that is, tables or indexes) for which the additional number of extents you specified would cause the corresponding tablespace to overflow.

Result

You have now checked the amount of freespace in the tablespace or tablespaces. See [Preventing Tablespace Overflow \[Page 233\]](#) for information on how to interpret the results to prevent tablespace overflow.



Preventing Tablespace Overflow

Use

This section describes how to prevent tablespace overflow in your Oracle database.

Prerequisites

You have completed the following procedures:

- [Checking Freespace for Tablespaces \[Page 231\]](#)
- [Checking Freespace for Objects in Tablespace \[Page 232\]](#)

This enables you to identify tablespaces that are about to overflow, and the objects that would cause overflow if they were extended.

Procedure

1. If an overflow has already occurred, you can check this in the SAP System log.
The Oracle error is given, together with the table or index that was accessed when the overflow occurred.
You can then determine which tablespace the table or index belongs to, using SAPDBA. For more information, see [Checking Freespace for Tablespaces \[Page 231\]](#). Set the number of extents to 1 when running the report.
2. If you have found that the tablespace would overflow if:
 - An object were to be extended by **one** extent, you must [extend the tablespace \[Page 233\]](#) as soon as possible.
The exception to this rule is for tablespaces `PSAPPROTD` or `PSAPPROTI`, which contain SAP logs from background processes and spool requests. Delete old log files and requests instead of extending these tablespaces.
 - An object were to be extended by **two** extents, you need to closely monitor the tablespace.
3. If you have found that there is no danger of overflow if two extents were added, the tablespace currently has sufficient free storage space. You need take no action at present, provided the tablespace continues to be used normally. Continue to monitor the tablespace regularly as usual.



If you change the storage parameters for a tablespace, or perform operations in the SAP System that cause it to fill more quickly than usual, be sure to monitor the tablespace closely and extend it if necessary.

24.4.2 Extending a Tablespace

Use

24.4 Tablespace Administration with SAPDBA

You can use SAPDBA for Oracle to extend a tablespace, that is, add a data file. The SAP System can be either active or closed when you extend a tablespace.

You need **not** shut down the database system if you want to add a data file to a tablespace.



Instead of adding a new data file to extend a tablespace, you can also use SAPDBA to:

- [Resize existing data files \[Page 275\]](#).
- [Switch on the AUTOEXTEND option for existing data files \[Page 192\]](#)

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- SAPDBA only supports the addition of data files that correspond to the SAP naming conventions. For more information, see [SAP Naming Conventions for Tablespaces and Data Files \[Page 49\]](#).

Procedure

1. Choose *Tablespace Administration* → *Tablespace* and enter the name of the tablespace that you want to extend by adding a data file.
2. Choose *Alter tablespace <tablespace name> add data file*.

SAPDBA shows a new menu for altering the tablespace.

SAPDBA proposes a path and size for the new data file. It also displays the details of the file system or raw device and its free space, as in the following example for a file system.



```
New path: '/oracle/LMA/sapdata1/btabd_2/btabd.data2'
New size: 121624 K
File system: '/dev/dsk/c1t1d0s2'
Free space: 121624 K
```

3. To show the details of existing data files, choose *Show existing data files*.

SAPDBA displays the details of the file (or files), including its sequence number, size, and full name with path. For example:

```
# Size in K Data file name (file system)
```

```
-----
```

```
1 335656 /oracle/LMA/sapdata1/btabd_1/btabd.data1
(/dev/dsk/c1t1d0s2)
```

4. To change the path for the new data file, choose *New path*, *Select new path from a list of file systems*, or *Select new path from a list of raw devices*.

You can only change the path, not the data file name. In the above example, you could change `/oracle/LMA/sapdata`.

For more information, see [Specifying the Path for the New Data File \[Page 238\]](#).


5. To change the size for the new data file, choose *New Size* and enter the size in bytes, kilobytes, megabytes, gigabytes, or terabytes.

For more information, see [Specifying the Size for the New Data File \[Page 239\]](#).

6. To add more than one new data file, do the following:

24.4 Tablespace Administration with SAPDBA

- a. Choose *Define more than one new data file* and select the next free entry.
SAPDBA displays default values for the new data file, as shown above for the first new data file.
 - b. Accept the default values or change them as required.
7. To set *Autoextend* from the default *OFF* to *ON*, do the following:
 - a. Choose *Autoextend*.
SAPDBA prompts you for the incremental and maximum file sizes used in automatic table extension:
 - The incremental size is the size of the new file that is automatically added to the tablespace to prevent overflow.
 - The maximum size is the maximum size for automatic extension of the tablespace. Beyond the maximum size, you have to extend the tablespace manually.
 - b. Enter the incremental and maximum file sizes.
 8. Choose *Start* to begin adding the data file.
SAPDBA checks that there is enough space to add the new data file, showing a display like the following:



```

SAPDBA: Checking space for data files of tablespace
-----
File system       :    '/dev/dsk/c1t1d0s2'
Current free space :    +      20856 K
New data files    :    -         56 K
Left free space   :    = +      20800 K    -> O.K.
-----

SAPDBA also performs some other checks before adding the file. For more
information, see Checks for Tablespace Extension and Creation \[Page 240\].
  
```

9. Enter RETURN to add the data file(s).
SAPDBA adds the new data file(s) to extend the tablespace and also generates a log, `<timestamp>.ext`. You can check this log to verify that the tablespace extension was successful.
SAPDBA creates a directory or a symbolic link for the data file. The control files (old and new) are backed up under the directory `<timestamp>` belonging to the log file `<timestamp>.ext`. SAPDBA then creates the data file with `ALTER TABLESPACE ADD DATAFILE`. If any of these operations fail, SAPDBA stops immediately.



If SAPDBA cannot add the new data file, it deletes the corresponding directory or symbolic link and cancels the tablespace extension. Your system and database are unchanged. Use the Oracle ALERT and trace files to analyze and correct the problem, then repeat the tablespace extension.

When SAPDBA has successfully created the new data file, it automatically goes to the *Backup database* menu, so that you can back up the extended tablespace.

10. Be sure to back up the extended tablespace. For more information, see [Backing Up After Tablespace Operations \[Page 241\]](#).

24.4.3 Creating or Dropping a Tablespace

24.4 Tablespace Administration with SAPDBA

Use

You can use SAPDBA for Oracle to create or drop a tablespace. The SAP System can be either active or closed when you create or drop a tablespace.

You need **not** shut down the database system if you want to create or drop a tablespace.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- SAPDBA only supports the creation of tablespaces that correspond to the naming conventions for SAP tablespaces (PSAP<name>). You can create the tablespaces PSAPROLL and PSAPTEMP with SAPDBA, but not SYSTEM. However, you can [extend \[Page 233\]](#) the SYSTEM tablespace.
For more information, see [SAP Naming Conventions for Tablespaces and Data Files \[Page 49\]](#).
- When you drop a tablespace, SAPDBA first checks that it is empty. You cannot drop a tablespace that is not empty. Do **not** normally delete the tablespace PSAPTEMP: if you need to do so for exceptional reasons, you must stop the SAP System first.

Procedure

Creating a Tablespace

1. Choose *Tablespace Administration* → *Tablespace* and enter the name of the tablespace that you want to create.
SAPDBA validates the name you choose, making sure that it conforms to SAP naming conventions.
2. Choose *Create tablespace <tablespace name>*.

SAPDBA shows a new menu for creating the tablespace.

SAPDBA proposes a path and size for the new data file. It also displays the details of the file system or raw device and its free space, as in the following example for a file system:



```
New path: '/oracle/LMA/sapdata1/btabd_2/btabd.data2'
New size: 121624 K
File system: '/dev/dsk/c1t1d0s2'
Free space: 121624 K
```

3. To change the path of the new file, choose *New path*, *Select new path from a list of file systems*, or *Select new path from a list of raw devices*.

You can only change the path, not the data file name. In the above example, you could change /oracle/LMA/sapdata.

For more information, see [Specifying the Path for the New Data File \[Page 238\]](#).

4. To change the size for the new data file, choose *New Size* and enter the size in bytes, kilobytes, megabytes, gigabytes, or terabytes.

For more information, see [Specifying the Size for the New Data File \[Page 239\]](#).

5. To define more than one new data file, do the following:
 - a. Choose *Define more than one new data file* and select the next free entry.

24.4 Tablespace Administration with SAPDBA

SAPDBA displays default values for the new data file, as shown above for the first new data file.

- b. Accept the default values or change them as required.
6. To set *Autoextend* from the default *OFF* to *ON*, do the following:
 - a. Choose *Autoextend*.

SAPDBA prompts you for the incremental and maximum file sizes used in automatic table extension:

- The incremental size is the size of the new file that is automatically added to the tablespace to prevent overflow.
- The maximum size is the maximum size for automatic extension of the tablespace. Beyond the maximum size, you have to extend the tablespace manually.

- b. Enter the incremental and maximum file sizes.
7. To change the default storage parameters, do the following:
 - a. Choose *Default Storage Parameters*.

SAPDBA displays the default values, such as in the following example:

a -	INITIAL	16 K
b -	NEXT	40 K
c -	MINEXTENTS	1
d -	MAXEXTENTS	100
e -	PCTINCREASE	0

For more information on the parameters, see the Oracle documentation.

- b. Change the default values as required.
8. Choose *Start* to begin adding the data file.

SAPDBA checks that there is enough space to add the new data file, showing a display like the following:



```
SAPDBA: Checking space for data file of tablespace
-----
File system       :    '/dev/dsk/c1t1d0s2'
Current free space :    +      20792 K
New data files    :    -      1600 K
Left free space   :    = +      19192 K    -> O.K.
-----
SAPDBA also performs some other checks before adding the file. For more
information, see Checks for Tablespace Extension and Creation \[Page 240\].
```

9. Enter RETURN to add the data file(s).

SAPDBA adds the new data file(s) to create the tablespace and also generates a log, `<timestamp>.ext`. You can check this log to verify that the tablespace creation was successful.

SAPDBA creates a directory or a symbolic link for the data file. The control files (old and new) are backed up under the directory `<timestamp>` belonging to the log file

24.4 Tablespace Administration with SAPDBA

<timestamp>.ext. SAPDBA then creates the data file using `CREATE TABLESPACE WITH <tablespace details>`. If any of these operations fail, SAPDBA stops immediately.



If SAPDBA cannot add the new data file, it deletes the corresponding directory or symbolic link and cancels the tablespace creation. Your system and database are unchanged. Use the Oracle ALERT and trace files to analyze and correct the problem, then repeat the tablespace creation.

When SAPDBA has successfully created the new data file, it automatically goes to the *Backup database* menu, so that you can back up the new tablespace.

10. Be sure to back up the new tablespace. For more information, see [Backing Up After Tablespace Operations \[Page 241\]](#).

Dropping a Tablespace

1. Choose *Tablespace Administration* → *Tablespace* and enter the name of the tablespace that you want to drop.
2. Choose *Drop tablespace <tablespace name>*.

SAPDBA displays information about the tablespace, such as in the following example:

```
Tablespace PSAPBTABD consists of
3 files (total size 436384 K)
```

3. Enter `y` to drop the tablespace.

SAPDBA checks that the tablespace is empty and drops it.



If SAPDBA cannot drop the tablespace, use the Oracle ALERT and trace files to analyze and correct the problem, then repeat the tablespace drop.

4. Restart the database to be sure of reclaiming the space occupied by the dropped tablespace.

24.4.4 Specifying the Path for the New Data File

Use

When you are using SAPDBA for Oracle to [extend an existing tablespace \[Page 233\]](#) or [create a new tablespace \[Page 235\]](#) – both of which involve adding a data file – you can specify a different path for the new data file. SAPDBA suggests a path for the data file, but this section tells you how to specify a different path if required.

Prerequisites

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.



We generally recommend for performance reasons that you store data files and redo log files on different disks. For more information, see [Organizing Disk Storage \[Page 46\]](#).

The following example shows how SAPDBA specifies the name of the data file:

24.4 Tablespace Administration with SAPDBA



You are adding a third data file to tablespace PSAPDOCUD in the SAP database C11. PSAPDOCUD currently 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`.

Therefore, the complete new file name is:

```
/oracle/C11/sapdata2/docud_3/docud.data3.
```

SAPDBA only allows you to change the directory path, `/oracle/C11/sapdata2/` in the above example. You cannot change the final part of the name, `/docud_3/docud.data3` in the above example.

Procedure

1. If you are working with file systems, choose a new path in one of the following ways:

- Using a different 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 other than the one suggested by SAPDBA.

SAPDBA creates the new data file in the directory that you entered. It does not create a symbolic link from the directory it recommended.

Note that specifying a different directory might result in competing I/O accesses, which can reduce the performance of your system.

- Using a non-standard path

You can create the file under a path which does not correspond to the SAP standard.

For example, you have set up and mounted a disk for data files. Therefore, new files must be added under path `/usr/newdisk` and not under the standard path.

SAPDBA then 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 (for example, `/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.

2. If you are working with raw devices, SAPDBA behaves as described above in the section “Using a non-standard path”. The symbolic link is set from `<SAPDATA_HOME>/sapraw` to the corresponding raw device. The file `<tablespace-name>.data<file number>` does not have to be created. See also [Raw Devices and SAPDBA \[Page 33\]](#).

24.4.5 Specifying the Size for the New Data File

Use

24.4 Tablespace Administration with SAPDBA

When you are using SAPDBA for Oracle to [extend an existing tablespace \[Page 233\]](#) or [create a new tablespace \[Page 235\]](#) – both of which involve adding a data file – you can specify a size for the new data file. SAPDBA suggests a size for the data file, but this section tells you how to specify a different size if required.

Prerequisites

SAPDBA calculates the suggested size of the new data file from the previous size of the tablespace using one of the following percentages:

- Percentage from the SAPDBA profile (`init<DBSID>.dba`) if the parameter [tspadd tspname \[Page 203\]](#) is set for this tablespace.
- Default percentage, which is 30 percent of the tablespace size when no entry exists for this tablespace in the SAPDBA profile.

For example, you might want to specify a different size from the above values if you know that the tablespace might grow more rapidly in future.

Procedure

1. Make a new data file large enough to accommodate the growth of the tablespace for a long period. For a critical, rapidly growing tablespace, make the new file 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 might later need to reorganize the tablespace. This can be extremely time-consuming. Also consider that you have to shut down the SAP 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, always add a sufficiently large file when extending a tablespace.

2. If possible, store I/O-intensive tablespaces each on separate hard disks.
3. 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 number of data files for the `SYSTEM` tablespace cannot be reduced after you have extended it. Instead, you can use SAPDBA to [resize data files \[Page 275\]](#), either reducing or increasing the file size.

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 make sure that all other users are logged off).

You can undo a tablespace extension to `PSAPROLL` only by means of irregular procedures using Oracle functions. A thorough knowledge of Oracle procedures is required.

24.4.6 Checks for Tablespace Extension and Creation

When you use SAPDBA for Oracle to [extend an existing tablespace \[Page 233\]](#) or [create a new tablespace \[Page 235\]](#) – both of which involve adding a data file – SAPDBA performs certain checks, which are described in this section.

Before an additional file is added, SAPDBA performs the following checks (if file systems are used):

24.4 Tablespace Administration with SAPDBA

- The following block sizes ought to be identical:

Block size	Used in
Filesystem block size	File system
Filesystem fragment size	File system fragments
Oracle block size	Database system

SAPDBA verifies that the above values are identical. If not, it issues a warning message and suggests selecting a new file system. We recommend that you ignore the warning because previous problems have been solved in current operating systems.

Also note the following:

- The Oracle database block size is always 8 KB in the SAP System.
- SAPDBA does not perform the check under the AIX operating system, as the block sizes are always different:

Block size for	Block Size
Oracle	4 KB
Filesystem	8 KB

For OSF1, the value for “filesystem fragment size” is always 1 KB, and so SAPDBA does not check it.

- 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 adds the data file to extend the tablespace or create a new tablespace.

24.4.7 Backing Up After Tablespace Operations

Use

When you have used SAPDBA for Oracle to extend a tablespace or create a new tablespace, you have changed the structure of the database. The database can be restored more quickly when you back up the new data file or the new tablespace and the control file.

For more information, see [Database Backup with SAPDBA \[Page 303\]](#).

Prerequisites

You have completed one of the following procedures:

- [Extending a Tablespace \[Page 233\]](#)
- [Creating or Dropping a Tablespace \[Page 235\]](#) (a backup is not necessary after dropping a tablespace)

Procedure

- Back up a newly created tablespace with the SAP program BRBACKUP.
- Back up a tablespace that you have extended with BRBACKUP.

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.

24.5 Reorganization with SAPDBA

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 `<SAPDATA_HOME>/sapreorg/<timestamp>`. They are given the names `cntrl<SAPSID>.old` and `cntrl<SAPSID>.new`, respectively.



If you extend several tablespaces in a row, do not back them up individually (this means that you have to exit the Backup menu), but instead perform a complete backup of the database once you have finished all the extensions (an online backup is sufficient but an offline backup is better).

24.4.8 Displaying File and Raw Device Information for a Tablespace

Use

You can use SAPDBA for Oracle to display the assignment of tablespaces to raw devices or data files.

Prerequisites

You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Choose *Tablespace Administration* → *Display all tablespaces and data files*.
SAPDBA displays the assignment tablespace to data files or tablespace to raw devices, and the size of the files.
2. Choose *Tablespace Administration* → *Display all raw device drivers in /dev*
In this display, which depends on your operating system, SAPDBA displays an overview of all raw devices and their size. If your database uses raw devices, the link to the raw device is also displayed.

For more information, see [Raw Devices and SAPDBA \[Page 33\]](#).

24.5 Reorganization with SAPDBA

Use

You can use SAPDBA for Oracle to reorganize the database. Reorganization improves the structure of the database, and can result in improved performance.

Prerequisites

When installed for an SAP System, the Oracle database looks as follows:

- Most of the tables and indexes of a tablespace are stored in only one extent.
- Each tablespace consists of exactly one data file.

This initial database status can change as follows:

- Additional extents

When more space for extra data is required, additional extents are created for the tables and indexes of a tablespace. The result might be poorer data access times.

24.5 Reorganization with SAPDBA

- Additional data files

When a tablespace is full – that is, there is not enough freespace to create a new extent – additional data files must be added (except if the AUTOEXTEND option is used).

- Freespace fragmentation

Adding or deleting complete objects causes freespace 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.

The Oracle system now automatically merges adjacent areas of free space, so this problem is less likely to occur than in the past.

- Internal fragmentation

This occurs if the fill level of the database blocks develops unevenly. The fill level of the individual blocks is initially identical. Creating and deleting data causes some blocks to be filled completely, while others remain relatively empty. As a result, space is used inefficiently.

- Block chaining

If a data record does not fit into a database block, block chaining occurs. When the record is accessed, the system 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 SAP Systems usually access table entries using an index, the above changes to the database do not normally increase the time required to access data. However, such changes can increase run times for full-table scans.

Effects of a Reorganization

A reorganization can have the following positive effects on the database:

- The data from one object is merged into a single extent.
- The data from one tablespace is merged into one or more data files.
- Freespace fragments in an object are merged into larger freespace segments, and this process is called "defragmentation".
- The fill level in the individual blocks is evened out, so reducing internal fragmentation.
- Data chains are resolved.

Features

- Reorganize various objects, such as, tables, indexes, lists of tables or indexes, tablespaces, tablespaces with data files
- Move or rename of the data files in a tablespace
- Resize the data files in a tablespace
- Restart or reset (that is, repeat from the start) an interrupted reorganization
- Change storage parameters, such as initial extent, for all reorganization types



SAPDBA supports reorganization of partitioned tables and indexes. The reorganization does **not** change the partitions and their parameters. If a partition of a partitioned table or index is in a tablespace that you want to reorganize, SAPDBA reorganizes all other partitions of the object in other

24.5 Reorganization with SAPDBA

tablespaces too, even if you do not specify that you want to reorganize the other tablespaces. In other words, SAPDBA reorganizes **all** partitions of a partitioned object.

SAPDBA also supports the reorganization of tables with all types of large object (LOB) columns. Large objects are recreated with the same physical characteristics as before the reorganization.

Activities

1. You [prepare for the reorganization \[Page 244\]](#), including deciding if it is really necessary.
2. If required, you [change storage parameters \[Page 253\]](#) without reorganizing.
3. If necessary, you [reorganize the database \[Page 259\]](#).

24.5.1 Preparations for Reorganization

Purpose

Before you start to reorganize your Oracle database using SAPDBA, you need to prepare properly.

Prerequisites

Before you start to prepare the reorganization, you consider the following:

- Scheduling the reorganization

It can take a long time to reorganize large quantities of data, depending on the speed of your hardware. Consider the following when scheduling the reorganization:

- Reorganize after starting the database

We recommend reorganizing immediately after opening the database in restricted session mode, because this means there are no locks on database objects.

- [Improve reorganization performance \[Page 248\]](#)

It is especially important to improve performance when [reorganizing certain tablespaces \[Page 270\]](#), such as PSAPPOOLD, PSAPSTABD, and PSAPBTABD.

If your host has several CPUs, you can save time by [reorganizing in parallel \[Page 248\]](#).

Also check the main log `reorg<DBSID>.log` to see whether a comparable operation has already run. This enables you to better estimate the time required.

- Run several reorganizations together

If possible, plan your reorganizations in advance, to enable you to perform them together. This minimizes disruptions to productive operation.

- Naming conventions

SAPDBA can only reorganize tablespaces for which the data files conform to the [SAP Naming Conventions for Tablespaces and Data Files \[Page 49\]](#). Be aware of the special features when working with [raw devices \[Page 33\]](#).

- Space

If you are reorganizing several objects specified in a list or tablespace, make sure that your file system (for the export/import method) or the affected tablespace (for the create as select method) have enough space for the objects to be reorganized. SAPDBA also checks this for you.

24.5 Reorganization with SAPDBA

If you want to reorganize a table or tablespace (without also reorganizing data files), make sure that enough free storage space is available in the relevant tablespace before the reorganization. Approximately 5% to 10% of the tablespace size is necessary. To check this, see:

- [Checking Freespace for Tablespaces \[Page 231\]](#)
- [Checking Freespace for Objects in Tablespace \[Page 232\]](#)



If your checks have revealed a large amount of free space in the objects, you can reorganize so that the storage parameters are adapted, or you can just [change the storage parameters \[Page 253\]](#) without also reorganizing.

Process Flow

1. You perform the following procedures to make sure that a reorganization is necessary:
 - a. You [check database statistics \[Page 245\]](#).
 - b. You [check tablespaces \[Page 230\]](#).
 - c. You [check extents and fragmentation \[Page 246\]](#).
 - d. You [decide if a reorganization is necessary \[Page 247\]](#).
2. If necessary, [you maximize the performance of the reorganization \[Page 248\]](#).
3. You decide whether to [reorganize in NOARCHIVELOG mode \[Page 249\]](#).
4. You [choose a reorganization type \[Page 251\]](#).
5. You shut down the SAP System.
6. You [switch the database to restricted session mode \[Page 221\]](#).
7. If necessary, you [back up the database \[Page 252\]](#).

Result

Now you are ready to start the [reorganization \[Page 259\]](#).



Checking Statistics for a Reorganization

Use

Before you reorganize the Oracle database with SAPDBA, we recommend that you check the database statistics for the objects you intend to reorganize. This helps you to [decide if a reorganization is necessary \[Page 247\]](#).

Procedure

Use the following to check statistics relevant to a reorganization:

- Computing Center Management System (CCMS):
 - [Monitoring Table and Index Fragmentation \(Oracle\) \[Ext.\]](#)
 - [Checking Storage Parameters \(Oracle\) \[Ext.\]](#)
- SAPDBA command line option [sapdba -check \[Page 347\]](#)
- SAPDBA command line option [sapdba -analyze \[Page 345\]](#)
- [Checking Extents and Fragmentation \[Page 246\]](#) with SAPDBA

24.5 Reorganization with SAPDBA

- [Check Tablespaces \[Page 230\]](#) with SAPDBA for information on data files, especially:
 - [Checking Freespace for Tablespaces \[Page 231\]](#)
 - [Extending a Tablespace \[Page 233\]](#)
 - [Displaying File and Raw Device Information for a Tablespace \[Page 242\]](#)
- Refer to [Reorganization of Special Tablespaces \[Page 270\]](#) for information on particular tablespaces.



Checking Extents and Fragmentation

Use

Before you reorganize the Oracle database with SAPDBA, we recommend that you check extents and fragmentation for the objects you intend to reorganize. This helps you to [decide if a reorganization is necessary \[Page 247\]](#).

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- For more information on logging checks, see [Setting the Log Checks Option \[Page 187\]](#) and [Log Files and Their Contents \[Page 205\]](#).

Procedure

1. Choose *Reorganization* → *Check extents and fragmentation*.
2. Choose *Objects with at least n extents* to display objects that might need to be reorganized due to having a large number of extents.
 The default is to check all objects in the database, displaying all that have more than two extents. You can refine the search by entering:
 - Criteria such as tablespace name
 - A different threshold for number of extents
3. Choose *Tablespace fragmentation (summary)* to display summary information on the size and fragmentation level of the specified tablespace.

To refine the display, enter a tablespace name.

Look at the following values:

- *Average number of allocated extents per segment*
- *Average number of free fragments per data file*

If these values are high, a reorganization is indicated. A value close to 1 is optimal.

4. Choose *Tablespace fragmentation (all extents)* to display detailed information on the size and fragmentation level of the selected tablespace.

SAPDBA displays the extents in the table according to their physical sequence within the data files.

Choose *Segment*, then:

- To see information for a single segment, enter its name.
- To see information on freespace at segment level, enter `free`.
- To see information on all extents, enter `all`.

24.5 Reorganization with SAPDBA

5. Choose one of the following to contrast allocated and used space in database objects:

- *Estimate/compute table*
- *Validate index*
- *Estimate/compute tablespace*



Estimate/Compute table/tablespace locks only the table record just examined.
Validate index locks the entire table belonging to the index.

These options are equivalent to the command `sapdba -analyze`. For more information on the parameters, see [sapdba -analyze \[Page 345\]](#).

You can use the information displayed as a starting point for reducing storage sizes of tables, indexes, or tablespaces during a reorganization. We recommend that you always leave a safety margin of 10-20% extra space in an object. This safety margin is automatically set when you reorganize with the *Reduce object size: yes* option.



Deciding if a Reorganization is Necessary

Use

Reorganization of an Oracle database is not required as often as you might think.

Prerequisites

You have completed:

- [Checking Statistics for a Reorganization \[Page 245\]](#)
- [Checking Extents and Fragmentation \[Page 246\]](#)

Procedure

Consider the following when deciding if a reorganization is necessary:

- Number of data files
 A reorganization including 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
 The SAP System normally uses indexes to access records. This means that full table scans are only rarely required. Therefore, it is not necessary to limit the number of extents per table as long as Oracle is capable of managing that number.
 Only consider a reorganization when total system performance declines due to a table or an index with a large number of extents. If many of the objects consist of a large number of extents, reorganize the relevant table, index, or 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 becomes necessary due to freespace problems.
 However, if none of the remaining fragments is large enough to create a new extent, [extend the tablespace \[Page 233\]](#) and possibly schedule a tablespace reorganization at a later date.

24.5 Reorganization with SAPDBA



Performance Improvements for Reorganization

Purpose

This section tells you how to improve the performance when you reorganize the Oracle database with SAPDBA. For more information, see the online help of SAPDBA and the SAP Notes database. Improving performance is particularly important during [Reorganization of Special Tablespaces \[Page 270\]](#).

Process Flow

1. You [change the initialization profile \[Page 248\]](#).
2. You [perform processes in parallel \[Page 248\]](#).



Changing the archiving mode for a reorganization does not significantly improve performance. For more information, see [Reorganizing in NOARCHIVELOG Mode \[Page 249\]](#).



Changing the Initialization Profile to Speed Up Reorganization

Use

You can [improve the performance of an Oracle reorganization \[Page 248\]](#) with SAPDBA by changing certain parameter values in the [initialization profile init<DBSID>.ora \[Page 47\]](#) before starting the reorganization.

Prerequisites

- Only make these changes for the duration of the reorganization.
- Stop the SAP System before starting this procedure.

Procedure

1. Write down the original values of parameters `sort_area_size` and `db_file_multiblock_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 [restricted session mode \[Page 221\]](#).
Now you can perform the reorganization.
5. **After** the reorganization, return the parameters to their original values before you start up the database and the SAP System for production use.



Performing Processes in Parallel to Speed Up Reorganization

Use

24.5 Reorganization with SAPDBA

You can [improve the performance of an Oracle reorganization \[Page 248\]](#) with SAPDBA by performing certain operations in parallel.



Be aware that, if you set the degree of parallelism to a value greater than 1, Oracle creates objects with the corresponding number of extents. SAPDBA divides the extent size for the reorganized table by the number of extents to be created, so that the overall size is as expected.

Procedure

Perform the following operations in parallel:

- Data transport, if you are using the [parallel export and import \[Page 289\]](#) method of reorganization

The export or import is performed in parallel by:

- Setting the degree of parallelism to greater than 1 using one of the following:
 - The option *Parallel* in the SAPDBA menu for *Export/import*.
 - The profile parameter [exp_imp_degree \[Page 201\]](#).
- Distributing the tables to be exported to several different export processes

In SAPDBA, define multiple export dump destinations. To export in parallel to a single dump destination, you can enter the same destination multiple times.

A corresponding number of export dump files is generated. The different export processes can be processed either sequentially or in parallel.

The time taken for the data export or import using the Oracle tools Export or Import depends on the amount of data. These procedures take a large part of the total processing time within the reorganization process. Therefore, we recommend you to perform the export or import in parallel.

For more information, see the Oracle documentation.

- Creating objects, if you are using the *Create As Select* reorganization method using SQL statements (that is, `CREATE TABLE AS SELECT`) for tables

The creation of large tables and their indexes can take a long time. Therefore, we recommend you to run these operations in parallel. For more information, see 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` (this is also for tables, not just indexes). For more information, see [index_degree \[Page 201\]](#).
- Set the degree of parallelism by setting the parameters *Parallel*, *ParTable*, or *ParIndex* in the SAPDBA reorganization menu.



It does not make sense to perform the specified processes in parallel if you select a degree of parallelism that is significantly larger than the number of CPUs in your system. Make sure that the degree of parallelism is no larger than two or three times the number of CPUs in your system.



Reorganizing in NOARCHIVELOG Mode

24.5 Reorganization with SAPDBA

Use

This procedure tells you how to reorganize the Oracle database with SAPDBA in NOARCHIVELOG mode.



If you use the [export/import reorganization method \[Page 284\]](#), a large number of redo log files are written. However, 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, which means that the CPU load is not significantly greater in ARCHIVELOG mode. Therefore, the reorganization is not significantly slowed down.

Prerequisites

Consider the following when deciding whether to run the reorganization in ARCHIVELOG or NOARCHIVELOG mode:

- NOARCHIVELOG mode has the following advantages:
 - You can safeguard against the risk of an archive stuck.
 - You save space in the archiving directory of the offline redo log files, `<SAPDATA_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, because you can resume from the backup performed immediately after the reorganization.
- We only recommend deactivating ARCHIVELOG mode under certain conditions, since it interrupts the continuity of the redo log files. Once the reorganization is complete, you **must** reactivate the ARCHIVELOG mode and perform a complete offline backup of the database. Remember that online backups are only secure **after** a complete offline backup of the database.
- Consider using NOARCHIVELOG mode when [reorganizing special tablespaces \[Page 270\]](#).

Procedure

1. For maximum security, perform an [online or offline backup of the entire database \[Page 252\]](#).

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 switching of the redo log file. 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 backed up to the archiving directory.

2. [Shut down the database \[Page 219\]](#) using SAPDBA.
3. [Deactivate ARCHIVELOG mode \[Page 41\]](#) using SAPDBA.
4. [Perform the reorganization \[Page 259\]](#).
5. After the reorganization, you **must immediately** [reactivate ARCHIVELOG mode \[Page 41\]](#) using SAPDBA.

6. For maximum security, perform a complete [offline backup of the database \[Page 252\]](#) immediately.

If you try to recover the database with an old backup, this would only be possible to the state before the reorganization (remember that the reorganization was **not** performed in ARCHIVELOG mode).

Although an online backup is possible, it is risky if every data file was not backed up. If the database crashes before the online backup is complete, 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 to the database during the online backup would be lost and would have to be repeated manually after the reorganization.



Choosing a Reorganization Type

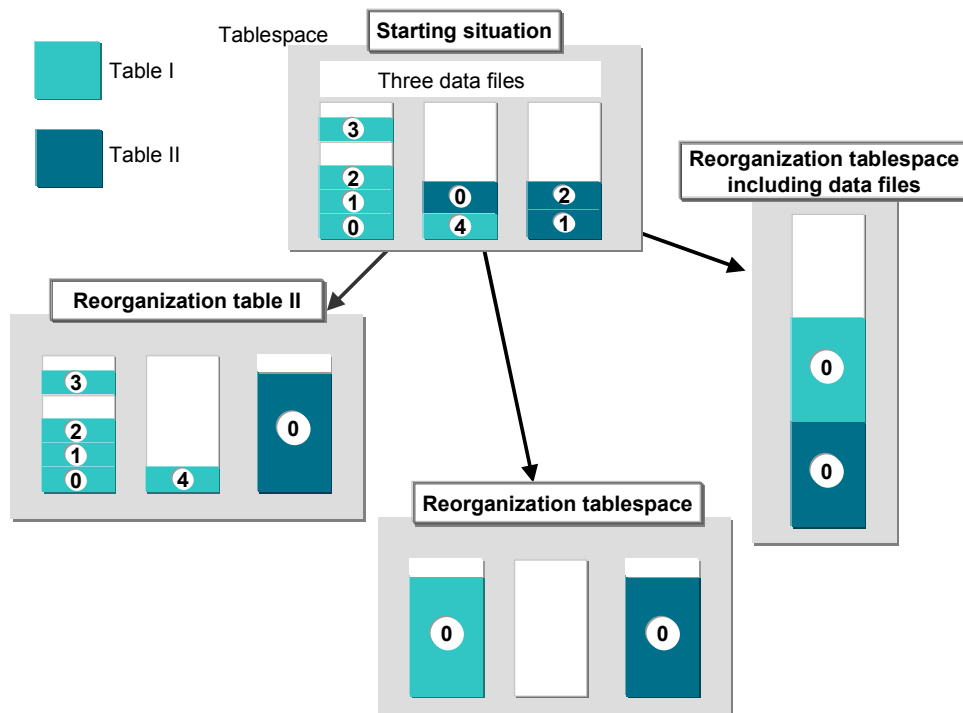
Use

There are several types of Oracle database reorganization. This section helps you choose the correct reorganization type. For example, if a specific table is causing space problems, you need only reorganize that table, not the entire tablespace.

Prerequisites

The following graphic is based on a tablespace consisting of three data files with two fragmented tables and fragmented freespace:

Main Reorganization Types for an Oracle Database



SAPDBA always defragments the tablespaces that are influenced by the reorganization. Once all the tables and indexes to be reorganized have been deleted, SAPDBA defragments the freespace of the tablespace, combining free fragments that are physically directly next to each other within all the data files. SAPDBA then recreates the deleted tables and indexes.

24.5 Reorganization with SAPDBA

Procedure

You choose the type of reorganization most appropriate to your needs:

- [Reorganization of a single object \[Page 260\]](#) (that is, table or index)

This type of reorganization merges the extents occupied by a table or index to one extent (using the default *compress extents: yes*). In addition, SAPDBA merges adjacent free storage fragments in the entire tablespace to form one closed area. The Oracle database system does this with its own system monitor.

Depending on the tablespace fragmentation, a table or index reorganization with the option *Compress extents: yes* might not be possible, even though the total freespace is larger than the table or index. SAPDBA can detect and warn you of this situation.

- [Reorganization of a list of objects \[Page 263\]](#) (that is, a list of tables and indexes)

In this type of reorganization SAPDBA reorganizes each object in the list file as described in the previous option, *Reorganization of a Single Object*.

- [Reorganization of a tablespace \[Page 267\]](#)

This type of reorganization merges all tables and all indexes of a tablespace to one extent per object. The freespace in the tablespace is merged into one fragment for each data file, but **not** when using the options *Create table ... as select* or *Index rebuild*.

- [Reorganization of a tablespace with data files \[Page 271\]](#)

This reorganization type is the most complicated and we recommend that you only perform it rarely. SAPDBA proceeds as follows:

- It performs the functionality described in the previous option *Reorganization of a Tablespace*.
- If the size of the resulting file is:
 - Smaller than 2 GB (this is the limit of most operating systems), SAPDBA merges the data files of a tablespace to one data file.
 - Larger than 2 GB, SAPDBA recommends writing the data to files smaller than 2 GB. SAPDBA writes all files to the same directory, following [SAP naming conventions \[Page 49\]](#).



Five files require 4.8 GB altogether. SAPDBA recommends creating three new data files with 1.6 GB each. You can change this recommended value, for example, to 2 x 1.9 GB and 1 x 1.0 GB, or 4 x 1.2 GB, and so on.

- [Moving or renaming the data files of a tablespace \[Page 274\]](#)

This procedure changes the data file allocation for a tablespace without actually reorganizing the tablespace.

- [Resizing the Data Files of a Tablespace \[Page 275\]](#)

This procedure increases or decreases the size of data files of the tablespace.



Backing Up the Database for Reorganization

Use

We **strongly recommend** that you back up the Oracle database before you reorganize it with SAPDBA. How you do this depends on the reorganization type. SAPDBA does **not**

24.5 Reorganization with SAPDBA

perform an automatic backup before or after a reorganization, although it sometimes displays the backup menu following a reorganization.

Procedure

Back up the database as follows:

- Reorganizing a single object or a list of objects

Before reorganizing a single object or list of objects, back up the data in the relevant table or tables, for example, using the [Export and Import \[Page 285\]](#) function in SAPDBA. If the reorganization fails and you cannot restart it, you can then use the backup of the data to restore the old status and repeat the reorganization.

- Reorganizing a tablespace with or without data files

- a. **Before** reorganizing a tablespace, make a complete backup of the database. It is best to perform an offline backup, although an online backup is also possible. Use BRBACKUP for the backup, which you can start [manually \[Page 73\]](#) or [with SAPDBA \[Page 303\]](#).

If the last complete backup of the database was performed recently, and all redo log files up to the start of the reorganization are available (on disk if possible), you can leave out the database backup. In this case, you could restore the database from the last available backup if necessary.

- b. **After** reorganizing a tablespace, you proceed as follows, depending on the reorganization type:

- i. Reorganization of a tablespace **without** data files

We recommend you to back up at least the reorganized tablespaces. This is because, in the event of database failure soon after the reorganization, you would have to repeat the reorganization process if there was no backup available, and the reorganization can take a long time.

- ii. Reorganization of a tablespace **with** data files

You **must** back up at least the reorganized tablespaces, because the underlying database structure has changed. Otherwise, SAPDBA cannot recover the database in the event of database failure. SAPDBA automatically displays the backup menu after the reorganization, to let you immediately start the backup.



A tablespace backup after reorganization is sufficient when the database is running in ARCHIVELOG mode. However, if it is running in [NOARCHIVELOG mode \[Page 249\]](#), you **must** back up the entire database. If many tablespaces require reorganization, we recommend a complete offline backup of the database before reorganization of the first tablespace and after reorganization of the last tablespace.

- Moving or renaming 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. This is because, in the event of database failure, SAPDBA cannot recover the database without a backup. In this case SAPDBA automatically displays the backup menu after the reorganization, to let you immediately start the backup.

24.5.2 Change Storage Parameters with SAPDBA

24.5 Reorganization with SAPDBA

Purpose

You can use SAPDBA for Oracle to change [storage parameters \[Page 256\]](#). Examples of storage parameters are the initial and next extents.

Prerequisites

Changing storage parameters is part of [Database Reorganization with SAPDBA \[Page 242\]](#).

Process Flow

You can change storage parameters in the following ways:

- During a [reorganization \[Page 259\]](#).
- During an [export \[Page 290\]](#).
- **Without** a reorganization or an export. See [Viewing and Changing Storage Parameters \[Page 254\]](#).



Viewing and Changing Storage Parameters

Use

You can view and change storage parameters using SAPDBA for Oracle:

- Directly from the reorganization menu using *Alter/show table or index storage parameters*.
- During a [reorganization \[Page 259\]](#).
- During an [export \[Page 290\]](#).

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- For more information on storage parameters, see [Storage Parameters \[Page 256\]](#).

Procedure

1. If you are performing this procedure directly from the reorganization menu, perform the following steps:
 - a. Choose *Reorganization* → *Alter/show table or index storage parameters*.
 - b. Enter *Owner* and *Table* names.
The owner is usually SAPR3. You can use a wildcard to find the table or index, for example, by entering ACT% to find all tables or indexes beginning with the letters ACT.
 - c. Select whether to use the ABAP dictionary to determine the size of the NEXT extent.
For more information, see [Default Value for NEXT \[Page 257\]](#).
 - d. Choose *Alter/show parameters* to view the current parameter values.
SAPDBA displays the current, suggested, and new values. Note that you can only change the values marked with an option on the left-hand side.

24.5 Reorganization with SAPDBA



We recommend that you only change the parameters `PCTINCREASE`, `PCTFREE`, `PCTUSED`, and `FREELISTS` in consultation with SAP, for example, if suggested by the [EarlyWatch \[Page 29\]](#) Service.

You cannot change all the parameters listed below if you are changing storage parameters directly from the reorganization menu.

2. Change the parameter values as required, considering the following factors:



When setting `INITIAL` and `NEXT`, it helps to check the space situation in the relevant tablespace. Refer to [Checking Freespace for Tablespaces \[Page 231\]](#).

– `INITIAL`

If you expect a table or index to grow quickly, it makes sense to increase `INITIAL`. The additional space in the initial 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.

You can reduce the initial extent if the table is static or shrinking in size. In this case, do **not** select the option to reduce the object size (*ReduceOb*) during a reorganization or export/import.

– `NEXT`

The suggested value for `NEXT` is calculated by SAPDBA. Refer to [Default Value for NEXT \[Page 257\]](#) for more information. Allow for the expected growth of the table or index when setting this value.

You can check for rapidly growing objects by [Checking Freespace for Objects in Tablespace \[Page 232\]](#).

If the table is growing rapidly, it makes sense to increase `NEXT`. To anticipate this situation, regularly use the SAPDBA command line option `sapdba -next`. See [sapdba -next <tablespace\(s\)> \[Page 351\]](#).

– `MINEXTENTS`

This is normally set to 1. It only needs to be greater than 1 if the table or index is so large that it does not fit into a single data file or the largest available freespace area.

– `MAXEXTENTS`

We recommend that `MAXEXTENTS` is set to 100, with a block size of 8 KB. This is the SAP default, that is, the soft limit. Starting with Oracle 7.3, you can set `MAXEXTENTS` to `UNLIMITED`. If `MAXEXTENTS` is set to zero, this means `UNLIMITED`.

`MAXEXTENTS` is automatically reset to the SAPDBA default value when there is an import or a reorganization of the relevant object. This value is set to the default value even when you use one of the options for changing storage parameters manually.



If the table is approaching the `MAXEXTENTS` limits as described in [Limitations of the Oracle Database System \[Page 36\]](#), you must reorganize it as soon as possible. If only the soft limit is reached, you can still increase `MAXEXTENTS`, but be sure to reorganize the table soon.

– `PCTFREE` and `PCTUSED`

24.5 Reorganization with SAPDBA

Only change these parameters in consultation with SAP or if you have extensive experience with Oracle database administration. You can simulate an asynchronous, linear PCTINCREASE response by using the command option `sapdba -next`. Refer to [sapdba -next <tablespace\(s\)> \[Page 351\]](#).

3. If you are performing this procedure for **multiple** objects – this is not possible if you are using *Alter/show table or index storage parameters* directly from the reorganization menu – select the following:
 - a. *next object*
Select this to set the parameters for the next object,
 - b. *name <%>*
Select this to restrict the display to objects beginning with a certain string. For example, enter ACT to view all objects beginning with the letters ACT.
 - c. *type*
Select this to restrict the display to tables, indexes, or all objects.

SAPDBA uses the settings from the previous menu to reorganize any objects for which you do not explicitly assign storage parameters
4. Choose *Commit* or *continue reorganization* to change the parameters to the values shown under *new value*.



Storage Parameters

Definition

The storage parameters for objects in an Oracle database such as tables or indexes define how the objects are stored.

Use

You can view and change the values of most storage parameters using SAPDBA.



We recommend that you only change the parameters PCTINCREASE, PCTFREE, PCTUSED, and FREELISTS in consultation with SAP, for example, if suggested by the [EarlyWatch \[Page 29\]](#) Service.

Structure

- INITIAL

This is the size of the single extent allocated to a table or an index when initially created. You can use SAPDBA to reduce the initial extent if required.

Specify a large value for the INITIAL extent of tables or indexes that grow quickly, allowing for future growth. This means that the database does not have to immediately assign a new extent.

Specify a smaller value for the INITIAL extent when you know that a table or index is unlikely to grow significantly.
- NEXT

This is the size of the second and all other extents assigned to a table or an index. This value is usually calculated using tables DD09L and TGORA/IGORA. See [Default Value for NEXT \[Page 257\]](#).

24.5 Reorganization with SAPDBA

You can generally leave the `NEXT` value set to the SAPDBA default value. However, if you have made large changes to the `INITIAL` value, increase or decrease the value of `NEXT` accordingly.

- `MINEXTENTS`

This is the initial number of extents when the table is created. It is normally set to 1. The exception is when 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`

This is the maximum number of extents allowed for the table. For more information, see [Limitations of the Oracle Database System \[Page 36\]](#).

- `PCTINCREASE`

This refers to the percentage increase in the `NEXT` extent size. It is always set to zero by SAPDBA and cannot be changed manually with SAPDBA. This prevents 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.

However, you can simulate an asynchronous, linear `PCTINCREASE` response by using the command option `sapdba -next`. Refer to [sapdba -next <tablespace\(s\)> \[Page 351\]](#).

- `PCTFREE`

This parameter specifies the percentage of storage in a data block that is held empty for possible changes to existing rows. The default value is 10%.

- `PCTUSED`

When a data block is full (except for the space for `PCTFREE`), no new rows are added to that block. No more rows can be added to that block until the percentage of used storage falls below the value of `PCTUSED`. The default value is 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.

- `FREELISTS`

This is the 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.



For values larger than 5 Oracle blocks, the values for `INITIAL` and `NEXT` are always rounded off to a multiple of 5 Oracle blocks. For example, if you define a `NEXT` value of 6 blocks, this is rounded off to 5. In further operations, the system uses a `NEXT` value of 10 Oracle blocks. This procedure is determined by Oracle and SAPDBA follows it. For more information, see [Default Value for NEXT \[Page 257\]](#).



Default Value for NEXT

Definition

SAPDBA for Oracle recommends a default value for the `NEXT` parameter when you do one of the following:

24.5 Reorganization with SAPDBA

- [Reorganize \[Page 259\]](#) a table
- [Export \[Page 290\]](#) a table
- [Change the storage parameters of a table \[Page 254\]](#)

For more information about the `NEXT` parameter, see [Storage Parameters \[Page 256\]](#).

Use

You can influence the calculation of the default value for extent parameter `NEXT` with the option *Use ABAP Dic. for NEXT*. The basis for calculating the `NEXT` parameters are the tables `DD09L` and `TGORA/IGORA` in the ABAP dictionary.

SAPDBA calculates a default value for the `NEXT` extent size as follows:

- *Use ABAP Dic. for NEXT: yes*

The new value for `NEXT` is calculated as the maximum of the following values:

- An SAP object is assigned a size category from table `DD09L`. The `NEXT` value is determined from table `TGORA/IGORA` based on that category. If the table does not have a category assigned in `DD09L`, this does not apply.
- Current `NEXT` value
- 10% of the total allocated space

SAPDBA compares the calculated maximum value with the values for the extent size in `TGORA/IGORA`. SAPDBA generally chooses the next bigger value found in `TGORA/IGORA` as the new `NEXT` value except if:

- The calculated maximum differs from the next smaller value of `TGORA/IGORA` by less than 5 Oracle blocks.

In this case, SAPDBA takes the smaller value. However, `NEXT` values that are larger than the largest `TGORA/IGORA` value are **not** reduced.

- `TGORA/IGORA` does not exist.

In this case, SAPDBA uses internal default values which correspond to the values in `TGORA/IGORA` (see "NEXT Values in `TGORA/IGORA`" below).

- *Use ABAP Dic. for NEXT: no*

SAPDBA does **not** use `DD09L` and `TGORA/IGORA` from the ABAP dictionary to derive the default value of `NEXT`. Instead, SAPDBA determines the maximum of the following two values:

- Current value for `NEXT`
- 10% of the total allocated space

Structure

Defining the `NEXT` values to correspond to the permanently assigned values in the `TGORA/IGORA` tables, as shown below, results in less wasted space in the tablespace. If the extent values are defined using a size category assigned in the ABAP Dictionary – see [Specification of SAP Tables in the ABAP Dictionary \[Page 25\]](#) – incorrectly defined values can be easily corrected in the database.

NEXT Values in `TGORA/IGORA`

Size	NEXT value	NEXT value for
------	------------	----------------

24.5 Reorganization with SAPDBA

category	for table (KB)	indexes (KB)
0	16	16
1	160	80
2	640	160
3	2560	640
4	10240	2560
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 note the following:

- 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. Therefore, consider carefully whether such large 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 40 KB) in order to avoid additional Oracle rounding. If you change values or insert new ones, it is best to specify only the corresponding multiple.

24.5.3 Reorganization

Purpose

This section tells you how to reorganize the Oracle database with SAPDBA.

Prerequisites

You have completed [Preparations for Reorganization \[Page 244\]](#).

Process Flow

You perform the [type of reorganization \[Page 251\]](#) that you identified during the preparations:

- [Reorganizing a single table or index \[Page 260\]](#)
- [Reorganizing a list of tables and indexes \[Page 263\]](#)
- [Reorganizing a tablespace \[Page 267\]](#)
- [Reorganizing a tablespace and data files \[Page 271\]](#)
- [Moving or renaming data files of a tablespace \[Page 274\]](#)

24.5 Reorganization with SAPDBA

- [Resizing the data files of a tablespace \[Page 275\]](#)
- [Restarting a reorganization \[Page 276\]](#)



Reorganizing a Single Table or Index

Use

You can use SAPDBA for Oracle to reorganize a single table or index.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Reorganization \[Page 244\]](#), especially [Backing Up the Database for Reorganization \[Page 252\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the reorganization in [Central Log reorg<DBSID>.log \[Page 207\]](#).
- SAPDBA supports online reorganization based on the Oracle internal PL/SQL functions.

Procedure

1. Choose *Reorganize single table or index*.
2. Enter *Owner* and *Table or index* names.

The owner is usually SAPR3. You can use a wildcard to find the table or index, for example, by entering ACT% to find all tables or indexes beginning with the letters ACT.

3. If you want to move the table or index to a new tablespace, enter *Tablespace* name, which must follow [SAP naming conventions \[Page 49\]](#).

SAPDBA checks the size available in the new tablespace and, if sufficient, displays the names of the original tablespace and the new tablespace. For example, PSAPBTABD → PSAPDOCUD indicates that you are moving a table between the tablespaces shown.

When you move a table, SAPDBA also moves the indexes to the corresponding tablespace, PSAPDOCUI in the above example.

4. Enter the *Working directory*.

This is the working directory for the reorganization. The default is <SAPDATA_HOME>/sapreorg or <SAPREORG>, based on the environment variable SAPREORG. Logs and scripts are saved in this directory. For more information on <SAPREORG>, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

5. If you are reorganizing a table, select one of the following [reorganization methods \[Page 284\]](#) for *Data Transfer*:
 - *Create As Select* method using SQL statements (that is, CREATE TABLE AS SELECT)
 - *Exp/Imp* method using Oracle export/import

24.5 Reorganization with SAPDBA



With *Online/offline reorganization* you can choose whether to reorganize the table online. Online reorganization is the default when the SAP System is running, otherwise offline reorganization is the default.

6. If you are using the export/import method for *Data Transfer*, enter the *Dump destination*.

This is the directory used to store the exported data. The default is specified by the [exireo_dumpdir \[Page 200\]](#) parameter. If it is not specified, then the default is <SAPDATA_HOME>/sapreorg.

Make sure that the space available in the directories or on the tapes is at least as large as the total size of all the objects to be reorganized. SAPDBA warns you if there is not enough space available. You can still continue with the reorganization even if SAPDBA issues a space warning.

If you are exporting to tape, SAPDBA asks you to state the size of the tape. SAPDBA also checks in this case whether the data fits on the tape. SAPDBA can only use one tape drive for each export dump file.



If SAPDBA uses the Oracle export program for the export to tape, write errors are possible that the Oracle export cannot recognize. Therefore, it is essential that you [back up \[Page 252\]](#) the corresponding data before starting such a reorganization. Otherwise you might lose data during the reorganization. In addition, always use the SAPDBA option *CheckExp: yes* (see parameters below). In general, an export to disk causes fewer errors than one to tape.

7. Select the required parameters for *Data transfer*:

- If you are reorganizing a **table**, SAPDBA displays the following parameters under *ORACLE exp/imp* or *Create As Select*:



SAPDBA only displays the parameters relevant to the reorganization **method** – that is, export/import or create as select – that you have chosen.

Parameter	Default	Method	Meaning
<i>ComprDmp</i>	NO	<i>Exp/Imp</i>	Compress the dump file, only possible for export to disk If selected, SAPDBA sends the data to the export dump files using the UNIX <code>compress</code> command. Do not use this parameter for tablespaces with objects already compressed by the SAP database interface, since it has no advantages (there might even be disadvantages).
<i>Chop</i>	NO	<i>Exp/Imp</i>	Chop the dump file, not possible for Windows platforms Select this if the export dump files are larger than the maximum file size (normally 2 GB) for your operating system. SAPDBA then sends the export data to a chop tool, which splits the data into several smaller files.
<i>CheckExp</i>	YES	<i>Exp/Imp</i>	Check dump files, recommended especially if exporting to tape SAPDBA performs a read check after the export using the <code>inx<TSP>.sql</code> scripts.

24.5 Reorganization with SAPDBA

<i>Commit</i>	YES	<i>Exp/Imp</i>	Commit command passed to database once buffer data has been imported
<i>ParTable</i>	1	<i>Create As Select</i>	Reorganize the table with or without parallel processing If you set this to 2 or more, SAPDBA recreates the table in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .
<i>Direct path export</i>	YES	<i>Exp/Imp</i>	Export data directly, without using SQL area This improves performance because the data is physically written straight to disk. In general, we recommend it.
<i>Buffer size</i>	3000000	<i>Exp/Imp</i>	To accelerate the reorganization, we recommend providing at least 3 MB of buffer space.

- If you are reorganizing an **index**, SAPDBA displays the following parameters under *Index type*:

NONUNIQUE or *UNIQUE*

NORMAL, *NORMAL/REV*, or *BITMAP*

If the index is non-unique, you can change the index type to any of the three types given above, that is *NORMAL*, *NORMAL/REV*, or *BITMAP*. However, if the index is unique, you cannot use *BITMAP*.

- Enter the required [storage parameters \[Page 296\]](#) under *Storage parameters*.
- Enter the required *Object handling* parameters:

Parameter	Default	Method	Meaning
<i>HideTab</i>	NO	Both	Hide the table, applies only to the reorganization of a single table If you set this to YES, all the tables that are reorganized or 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).
<i>DropTab</i>	<i>normal</i>	<i>Exp/Imp</i>	Drop the table The default is to drop the table after the data export. If you choose <i>defer</i> , the table is only dropped after the data has been imported. Although this improves data security during the reorganization, it means that enough space is required for two copies of the table (that is, the old and the new). If SAPDBA detects that there is not enough space, it automatically switches to <i>normal</i> mode.
<i>ParIndex</i>	1	–	Reorganize the index with or without parallel processing By increasing this to 2 or more, SAPDBA recreates the index in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .

- Choose *Start* to prepare for the reorganization.

SAPDBA prepares for the reorganization by:

24.5 Reorganization with SAPDBA

- Prompting to make sure that you [backed up \[Page 252\]](#) the database
 - Warning you if the SAP System is still running
 - [Checking for freespace \[Page 280\]](#)
 - Checking for space in the dump file if you are using the export/import method
 - Generating the:
 - <timestamp> subdirectory in the working directory
 - <timestamp>.rsi log
 - Scripts required to perform the reorganization
 - restart.rsi file for restart in the <timestamp> subdirectory
11. Reply to the prompts in the preparation phase as necessary.
Finally, SAPDBA prompts you to start the reorganization.
12. Choose how to start the reorganization:
- Immediately in the foreground
 - In the [background \[Page 283\]](#)
 - Not now, but possibly later

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the reorganization \[Page 276\]](#) after fixing the problem.

**Reorganizing a List of Tables and Indexes****Use**

You can use SAPDBA for Oracle to reorganize a list of tables or indexes.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Reorganization \[Page 244\]](#), especially [Backing Up the Database for Reorganization \[Page 252\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the reorganization in [Central Log reorg<DBSID>.log \[Page 207\]](#).

Procedure

6. Create an ASCII file in the working directory with the list of the objects to be reorganized:
- There must be only one object name on each line.
 - If required, you can enter the object owner **before** the object name.
 - SAPDBA assumes `sapr3` to be the default owner if there is only a single entry on a line.

24.5 Reorganization with SAPDBA

- Use a space or full stop to separate the object and owner.



Here is an example of a file containing a list of objects to be reorganized:

```
ACTFLI
ACTFLI~0
OWNIT ACTARC
```

The first two lines contain objects owned by `sapr3`. The final line contains an object owned by `OWNIT`.

2. Choose *Reorganize list of tables and indexes*.
3. Enter *List of tables/indexes*.

This is the name of the file that you created in the first step.

4. Enter the *Working directory*.

This is the working directory for the reorganization. The default is

`<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

5. Select one of the following [reorganization methods \[Page 284\]](#) for *Data Transfer*:
 - *Create As Select* method using SQL statements (that is, `CREATE TABLE AS SELECT`)
 - *Exp/Imp* method using Oracle export/import
6. If you are using the export/import method for *Data Transfer*, enter the *Dump destination*.

This is the directory used to store the exported data. The default is specified by the [exireo_dumpdir \[Page 200\]](#) parameter. If it is not specified, then the default is `<SAPDATA_HOME>/sapreorg`.

Make sure that the space available in the directories or on the tapes is at least as large as the total size of all the objects to be reorganized. SAPDBA warns you if there is not enough space available. You can still continue with the reorganization even if SAPDBA issues a space warning.

If you are exporting to tape, SAPDBA asks you to state the size of the tape. SAPDBA also checks in this case whether the data fits on the tape. SAPDBA can only use one tape drive for each export dump file.



If SAPDBA uses the Oracle export program for the export to tape, write errors are possible that the Oracle export cannot recognize. Therefore, it is essential that you [back up \[Page 252\]](#) the corresponding data before starting such a reorganization. Otherwise you might lose data during the reorganization. In addition, always use the SAPDBA option *CheckExp: yes* (see parameters below). In general, an export to disk causes fewer errors than one to tape.

7. Select the required parameters for *Data transfer*:



SAPDBA only displays the parameters relevant to the reorganization **method** – that is, export/import or create as select – that you have chosen.

Parameter	Default	Method	Meaning
-----------	---------	--------	---------

24.5 Reorganization with SAPDBA

<i>ComprDmp</i>	NO	<i>Exp/Imp</i>	Compress the dump file, only possible for export to disk If selected, SAPDBA sends the data to the export dump files using the UNIX <code>compress</code> command. Do not use this parameter for tablespaces with objects already compressed by the SAP database interface, since it has no advantages (there might even be disadvantages).
<i>Chop</i>	NO	<i>Exp/Imp</i>	Chop the dump file, not possible for Windows platforms Select this if the export dump files are larger than the maximum file size (normally 2 GB) for your operating system. SAPDBA sends the export data to a chop tool, which splits the data into several smaller files.
<i>CheckExp</i>	YES	<i>Exp/Imp</i>	Check dump files, recommended especially if exporting to tape SAPDBA performs a read check after the export using the <code>inx<TSP>.sql</code> scripts.
<i>Commit</i>	YES	<i>Exp/Imp</i>	Commit command passed to database once buffer data has been imported
<i>Parallel</i>	1	<i>Exp/Imp</i>	Reorganize with or without parallel processing By increasing this to 2 or more, SAPDBA recreates the tables in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .
<i>ParTable</i>	1	<i>Create As Select</i>	Reorganize with or without parallel processing By increasing this to 2 or more, SAPDBA recreates the tables in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .
<i>Direct path export</i>	YES	<i>Exp/Imp</i>	Export data directly, without using SQL area This improves performance because the data is physically written straight to disk. In general, we recommend it.
<i>Buffer size</i>	3000000	<i>Exp/Imp</i>	To accelerate the reorganization, we recommend providing at least 3 MB of buffer space.

8. Enter the required [storage parameters \[Page 296\]](#) under *Storage parameters*.

9. Enter the required *Object handling* parameters:

Parameter	Default	Method	Meaning
<i>HideTab</i>	NO	Both	Hide the table, applies only to the reorganization of a single table If you set this 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).

24.5 Reorganization with SAPDBA

<i>SAP-Tsp</i>	<i>NO</i>	Both	<p>Move objects to SAP-standard tablespaces</p> <p>If you set this to YES, SAPDBA places all objects to be reorganized in the tablespaces defined in the ABAP dictionary.</p>
<i>RebuildI</i>	<i>NO</i>	Both	<p>Use <code>ALTER INDEX REBUILD [ONLINE] SQL</code> statement</p> <p>If you set this to YES, the indexes to be reorganized are set up with the above SQL statement, not with <code>DROP/CREATE INDEX</code>. The <code>[ONLINE]</code> option is used automatically if possible.</p> <p>Advantages:</p> <ul style="list-style-type: none"> • Old index is available while new one is being created • Faster reorganization • Constraints need not be deleted <p>Disadvantages:</p> <ul style="list-style-type: none"> • More space required for extra copy of index • Freespace cannot always be optimally allocated • Can only be used when only indexes (not the accompanying tables) are being reorganized
<i>ParIndex</i>	<i>1</i>	–	<p>Reorganize indexes with or without parallel processing</p> <p>By increasing this to 2 or more, SAPDBA recreates indexes in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248].</p>

10. Choose *Start* to prepare for the reorganization.

SAPDBA prepares for the reorganization by:

- Prompting to make sure that you [backed up \[Page 252\]](#) the database
- Warning you if the SAP System is still running
- [Checking for freespace \[Page 280\]](#)
- Checking for space in the dump file if you are using the export/import method
- Generating the:
 - `<timestamp>` subdirectory in the working directory
 - `<timestamp>.rli` log
 - Scripts required to perform the reorganization
 - `restart.rli` file for restart in the `<timestamp>` subdirectory

13. Reply to the prompts in the preparation phase as necessary.

Finally, SAPDBA prompts you to start the reorganization.

14. Choose how to start the reorganization:

- Immediately in the foreground
- In the [background \[Page 283\]](#)

- Not now, but possibly later

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the reorganization \[Page 276\]](#) after fixing the problem.



Reorganizing a Tablespace

Use

You can use SAPDBA for Oracle to reorganize a tablespace.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Reorganization \[Page 244\]](#), especially [Backing Up the Database for Reorganization \[Page 252\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the reorganization in [Central Log reorg<DBSID>.log \[Page 207\]](#).
- If you are reorganizing the tablespaces PSAPPOOLD, PSAPSTABD, or PSAPBTABD, refer to [Reorganization of Special Tablespaces \[Page 270\]](#).



If you reorganize a tablespace containing tables according to the SAP naming conventions (that is, the name ends with “D” for data), SAPDBA always reorganizes the corresponding index tablespace (that is, with the ending “I” for index).

Procedure

6. Choose *Reorganize tablespace*.
7. Enter *Tablespace name*.

This is the name of the tablespace to be reorganized.

8. Enter the *Working directory*.

This is the working directory for the reorganization. The default is <SAPDATA_HOME>/sapreorg or <SAPREORG>, based on the environment variable SAPREORG. Logs and scripts are saved in this directory. For more information on <SAPREORG>, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

9. Select one of the following [reorganization methods \[Page 284\]](#) for *Data transfer*:
 - *Create As Select* method using SQL statements (that is, CREATE TABLE AS SELECT)
 - *Exp/Imp* method using Oracle export/import
5. If you are using the export/import method, enter the *Dump destination*.

24.5 Reorganization with SAPDBA

This is the directory used to store the exported data. The default is specified by the [exireo_dumpdir \[Page 200\]](#) parameter. If it is not specified, then the default is `<SAPDATA_HOME>/sapreorg`.

Make sure that the space available in the directories or on the tapes is at least as large as the total size of all the objects to be reorganized. SAPDBA warns you if there is not enough space available. You can still continue with the reorganization even if SAPDBA issues a space warning.

If you are exporting to tape, SAPDBA asks you to state the size of the tape. SAPDBA also checks in this case whether the data fits on the tape. SAPDBA can only use one tape drive for each export dump file.



If SAPDBA uses the Oracle export program for the export to tape, write errors are possible that the Oracle export cannot recognize. Therefore, it is essential that you [back up \[Page 252\]](#) the corresponding data before starting such a reorganization. Otherwise you might lose data during the reorganization. In addition, always use the SAPDBA option *CheckExp: yes* (see parameters below). In general, an export to disk causes fewer errors than one to tape.

6. Select the required parameters for *Data transfer*:



SAPDBA only displays the parameters relevant to the reorganization **method** – that is, export/import or create as select – that you have chosen.

Parameter	Default	Method	Meaning
<i>ComprDmp</i>	NO	<i>Exp/Imp</i>	Compress the dump file, only possible for export to disk If selected, SAPDBA sends the data to the export dump files using the UNIX <code>compress</code> command. Do not use this parameter for tablespaces with objects already compressed by the SAP database interface, since it has no advantages (there might even be disadvantages).
<i>Chop</i>	NO	<i>Exp/Imp</i>	Chop the dump file, not possible for Windows platforms Select this if the export dump files are larger than the maximum file size (normally 2 GB) for your operating system. SAPDBA sends the export data to a chop tool, which splits the data into several smaller files.
<i>CheckExp</i>	YES	<i>Exp/Imp</i>	Check dump files, recommended especially if exporting to tape SAPDBA performs a read check after the export using the <code>inx<TSP>.sql</code> scripts.
<i>Commit</i>	YES	<i>Exp/Imp</i>	Commit command passed to database once buffer data has been imported
<i>Parallel</i>	1	<i>Exp/Imp</i>	Reorganize with or without parallel processing By increasing this to 2 or more, SAPDBA recreates the tables in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .

24.5 Reorganization with SAPDBA

<i>ParTable</i>	1	<i>Create As Select</i>	Reorganize with or without parallel processing By increasing this to 2 or more, SAPDBA recreates the tables in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .
<i>Direct path export</i>	YES	<i>Exp/Imp</i>	Export data directly, without using SQL area This improves performance because the data is physically written straight to disk. In general, we recommend it.
<i>Buffer size</i>	3000000	<i>Exp/Imp</i>	To accelerate the reorganization, we recommend providing at least 3 MB of buffer space.

7. Enter the required [storage parameters \[Page 296\]](#) under *Storage parameters*.

8. Enter the required *Object handling* parameters:

Parameter	Default	Method	Meaning
<i>HideTab</i>	NO	Both	Hide the table, applies only to the reorganization of a single table If you set this 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).
<i>SAP-Tsp</i>	NO	Both	Move objects to SAP-standard tablespaces If you set this to YES, SAPDBA places all objects to be reorganized in the tablespaces defined in the ABAP dictionary.
<i>RebuildI</i>	NO	Both	Use ALTER INDEX REBUILD SQL statement If you set this to YES, the indexes to be reorganized are set up with the above SQL statement, not with DROP/CREATE INDEX. Advantages of ALTER INDEX REBUILD: <ul style="list-style-type: none"> • Old index is available while new one is being created • Faster reorganization • Constraints need not be deleted Disadvantages: <ul style="list-style-type: none"> • More space required for extra copy of index • Freespace cannot always be optimally allocated • Can only be used when only indexes (not the accompanying tables) are being reorganized
<i>ParIndex</i>	1	–	Reorganize indexes with or without parallel processing By increasing this to 2 or more, SAPDBA recreates indexes in parallel using the number of processes you enter. For more information, see Performing Processes in

24.5 Reorganization with SAPDBA

			Parallel to Speed Up Reorganization [Page 248] .
--	--	--	--

9. Choose *Start* to prepare for the reorganization.

SAPDBA prepares for the reorganization by:

- Prompting to make sure that you [backed up \[Page 252\]](#) the database
- Warning you if the SAP System is still running
- [Checking for freespace \[Page 280\]](#)
- Checking for space in the dump file if you are using the export/import method
- Generating the:
 - `<timestamp>` subdirectory in the working directory
 - `<timestamp>.rtc` log
 - Scripts required to perform the reorganization
 - `restart.rtc` file for restart in the `<timestamp>` subdirectory

15. Reply to the prompts in the preparation phase as necessary.

Finally, SAPDBA prompts you to start the reorganization.

16. Choose how to start the reorganization:

- Immediately in the foreground
- In the [background \[Page 283\]](#)
- Not now, but possibly later

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the reorganization \[Page 276\]](#) after fixing the problem.

After the reorganization, we strongly recommend you to [back up the database \[Page 252\]](#), or at least the reorganized tablespaces.



Reorganization of Special Tablespaces

When reorganizing the following tablespaces with SAPDBA for Oracle, you might run into problems:

- PSAPPOOLD

This tablespace contains the `ATAB` table with a very large number of records. The import – and therefore the reorganization – of this tablespace can take up to 10 hours, depending on your operating system. The run time 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. For more information, see the SAP Notes.

Also, if you operate a productive database in `ARCHIVELOG` mode, 1 to 2 GB of free space is required in the archiving directory of the offline redo log files (`<SAPDATA_HOME>/saparch`). Therefore, it helps to run the database in `NOARCHIVELOG` mode during the reorganization of PSAPPOOLD. If you reorganize in `NOARCHIVELOG` mode, be sure to back up the entire database afterwards. For more information, see [Reorganizing in NOARCHIVELOG Mode \[Page 249\]](#).

24.5 Reorganization with SAPDBA

- PSAPSTABD

This tablespace usually contains the most tables with the highest total number of extents, constraints, and so on. 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 large enough to avoid bottlenecks. For more information, see [Database Parameters \[Page 47\]](#).

- PSAPBTABD

This tablespace grows very quickly. To avoid frequent reorganization, we recommend regular use of the command `sapdba -next <tablespace(s)>` [\[Page 351\]](#).

Refer to [Performance Improvements for Reorganization \[Page 248\]](#) for general information about avoiding problems during the reorganization of these tablespaces. Use the SAP Notes for current information about improving performance.



Reorganizing a Tablespace and Data Files

Use

You can use SAPDBA for Oracle to reorganize a tablespace with data files. It is only possible to use the export/import reorganization method for this procedure. The procedure allows you to consolidate a tablespace into the minimum number of data files. This [type of reorganization \[Page 251\]](#) is the most complicated and we recommend that you only perform it occasionally.

The procedure has the following features:

- You can change the default values for file size and file directory. Therefore, you can optimize disk space utilization through appropriate distribution of the files.
- You can change the number of data files, their respective sizes, and their path. All data files must be smaller than 2 GB and at least one data file must be 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.



The 2 GB maximum reflects the file size limit of most operating systems.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Reorganization \[Page 244\]](#), especially [Backing Up the Database for Reorganization \[Page 252\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the reorganization in [Central Log reorg<DBSID>.log \[Page 207\]](#).
- Make sure that no Oracle background processes are running. Hanging background processes can disturb the tablespace reorganization because they keep data 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.

24.5 Reorganization with SAPDBA



If you reorganize a tablespace containing tables according to the SAP naming conventions (that is, the name ends with "D" for data), SAPDBA always reorganizes the corresponding index tablespace (that is, with the ending "I" for index).

Procedure

10. Choose *Reorganize tablespace and data files*.

11. Enter *Tablespace name*.

This is the name of the tablespace to be reorganized.

12. Enter the *Working directory*.

This is the working directory for the reorganization. The default is `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

13. Enter the *Dump destination*.

This is the directory used to store the exported data. The default is specified by the [exireo_dumpdir \[Page 200\]](#) parameter. If it is not specified, then the default is `<SAPDATA_HOME>/sapreorg`.

Make sure that the space available in the directories or on the tapes is at least as large as the total size of all the objects to be reorganized. SAPDBA warns you if there is not enough space available. You can still continue with the reorganization even if SAPDBA issues a space warning.

If you are exporting to tape, SAPDBA asks you to state the size of the tape. SAPDBA also checks in this case whether the data fits on the tape. SAPDBA can only use one tape drive for each export dump file.



If SAPDBA uses the Oracle program export for the export to tape, write errors are possible that the Oracle export cannot recognize. Therefore, it is essential that you [back up \[Page 252\]](#) the corresponding data before starting such a reorganization. Otherwise you might lose data during the reorganization. In addition, always use the SAPDBA option `CheckExp: yes` (see parameters below). In general, an export to disk is less error-prone than one to tape.

14. Select the required parameters for *ORACLE exp/imp*.

Parameter	Default	Meaning
<i>ComprDmp</i>	NO	Compress the dump file, only possible for export to disk If selected, SAPDBA sends the data to the export dump files using the UNIX <code>compress</code> command. Do not use this parameter for tablespaces with objects already compressed by the SAP database interface, since it has no advantages (there might even be disadvantages).
<i>Chop</i>	NO	Chop the dump file, not possible for Windows platforms Select this if the export dump files are larger than the maximum file size (normally 2 GB) for your operating system. SAPDBA sends the

24.5 Reorganization with SAPDBA

		export data to a chop tool, which splits the data into several smaller files.
<i>CheckExp</i>	YES	Check dump files, recommended especially if exporting to tape SAPDBA performs a read check after the export using the <code>inx<TSP>.sql</code> scripts.
<i>Commit</i>	YES	Commit command passed to database once buffer data has been imported
<i>Parallel</i>	1	Reorganize with or without parallel processing By increasing this to 2 or more, SAPDBA recreates the tables in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .
<i>Direct path export</i>	YES	Export data directly, without using SQL area This improves performance because the data is physically written straight to disk. In general, we recommend it.
<i>Buffer size</i>	3000000	To accelerate the reorganization, we recommend providing at least 3 MB of buffer space.

15. Enter the required [storage parameters \[Page 296\]](#) under *Storage parameters*.

16. Enter the required *Object handling* parameters:

Parameter	Default	Meaning
<i>HideTab</i>	NO	Hide the table, applies only to the reorganization of a single table If you set this 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).
<i>SAP-Tsp</i>	NO	Move objects to SAP-standard tablespaces If you set this to YES, SAPDBA places all objects to be reorganized in the tablespaces defined in the ABAP dictionary.
<i>RebuildFi</i>	NO	Reduce data file size If you set this to YES, SAPDBA reduces the file size during the reorganization. SAPDBA asks how much free space you want to leave, recommending 10% as the default.
<i>ParIndex</i>	1	Reorganize indexes with or without parallel processing By increasing this to 2 or more, SAPDBA recreates indexes in parallel using the number of processes you enter. For more information, see Performing Processes in Parallel to Speed Up Reorganization [Page 248] .

10. Choose *Start* to prepare for the reorganization.

SAPDBA prepares for the reorganization by:

- Prompting to make sure that you [backed up \[Page 252\]](#) the database
- Warning you if the SAP System is still running
- [Checking for freespace \[Page 280\]](#)

24.5 Reorganization with SAPDBA

- Checking for space in the dump file if you are using the export/import method
 - Generating the:
 - `<timestamp>` subdirectory in the working directory
 - `<timestamp>.rtd` log
 - Scripts required to perform the reorganization
 - `restart.rtd` file for restart in the `<timestamp>` subdirectory
17. Reply to the prompts in the preparation phase as necessary.
- Finally, SAPDBA prompts you to start the reorganization.
18. Choose how to start the reorganization:
- Immediately in the foreground
 - In the [background \[Page 283\]](#)
 - Not now, but possibly later

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the reorganization \[Page 276\]](#) after fixing the problem.



Moving or Renaming the Data Files of a Tablespace

Use

You can use SAPDBA for Oracle to move or rename the data files of a tablespace.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Reorganization \[Page 244\]](#), especially [Backing Up the Database for Reorganization \[Page 252\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the reorganization in [Central Log reorg<DBSID>.log \[Page 207\]](#).

Procedure

17. Choose *Reorganization* → *Move/rename data files of a tablespace*.

18. Enter *Tablespace* name.

This is the name of the tablespace for which you want to move or rename the data files.

19. Enter the *Working directory*.

This is the working directory for the reorganization. The default is `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

20. Choose *Start* to prepare for the reorganization.

SAPDBA prepares for the reorganization by:

- Prompting to make sure that you [backed up \[Page 252\]](#) the database
- Warning you if the SAP System is still running
- [Checking for freespace \[Page 280\]](#)
- Checking for space in the dump file if you are using the export/import method
- Generating the:
 - `<timestamp>` subdirectory in the working directory
 - `<timestamp>.rmv log`
 - Scripts required to perform the reorganization
 - `restart.rmv` file for restart in the `<timestamp>` subdirectory

19. Reply to the prompts in the preparation phase as necessary.

SAPDBA prompts you to change the data file specification.

20. Enter the number of the data file that you want to change and choose a new path on a file system or raw device.

SAPDBA performs the final checks and prompts you to start the reorganization.

21. Choose how to start the reorganization:

- Immediately in the foreground
- In the [background \[Page 283\]](#)
- Not now, but possibly later



If you chose the `SYSTEM` tablespace or a tablespace with rollback segments, SAPDBA closes the database instance with `shutdown immediate`. The instance is then brought to mount status. After the reorganization, SAPDBA opens the database.
For all other tablespaces, SAPDBA puts the tablespace offline but the database remains open. After the reorganization, SAPDBA puts the tablespace online.

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the reorganization \[Page 276\]](#) after fixing the problem.

After the reorganization, you **must** [back up the database \[Page 252\]](#). SAPDBA displays the backup menu on completion.



Resizing the Data Files of a Tablespace

Use

You can use SAPDBA for Oracle to change the sizes of the data files of a tablespace (including the `SYSTEM` tablespace), using the Oracle command `ALTER DATABASE`

24.5 Reorganization with SAPDBA

DATAFILE ... RESIZE... You can perform this procedure while the SAP System is running.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- SAPDBA logs the reorganization in [Central Log reorg<DBSID>.log \[Page 207\]](#).

Procedure

21. Choose *Reorganization* → *Resize data files of a tablespace*.

22. Enter *Tablespace* name.

This is the name of the tablespace for which you want to resize the data files.

23. Choose *Start*.

SAPDBA displays the data files of the selected tablespace.

24. Choose the data file to resize.

25. SAPDBA displays the current details for the data file.

26. Enter the *New size* for the data file.

SAPDBA automatically shows and checks the maximum file size according to the free space in the file system and the minimum file size according to the used extents inside the file.

27. If required, select *Autoextend*.

SAPDBA prompts you for the incremental and maximum file sizes used in automatic table extension:

- The incremental size is the size of the new file that is automatically added to the tablespace to prevent overflow.
- The maximum size is the maximum size for automatic extension of the tablespace. Beyond the maximum size, you have to extend the tablespace manually.

Enter the incremental and maximum file sizes.

For more information, see [Effects of Autoextend and Resize on SAPDBA \[Page 192\]](#).

28. Choose *Start and execute changes*.

No scripts are created for this reorganization type.

Result

SAPDBA resizes the data files immediately. It logs the changes in the file <timestamp>.rrs. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).



Restarting a Reorganization

Use

You can restart a reorganization with SAPDBA for Oracle:

- If the reorganization failed due to a problem in the main phase, but **not** if the problem was in the preparatory phase. You can restart it from the point of failure.

24.5 Reorganization with SAPDBA

- If you stopped the reorganization after the preparatory phase because you wanted to restart it later

In certain circumstances, you can reset the reorganization, which means setting the database back to how it was before the reorganization. We recommend that you **only** reset the reorganization if you do **not** want to restart it. SAPDBA only displays the *Reset* option in the reorganization menu if a reset is possible.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have identified the:
 - Log written by SAPDBA during the reorganization
You can use SAPDBA to do this as described in "Procedure" below or find it yourself on disk as follows (these are default locations):
 - UNIX: <SAPDATA_HOME>/sapreorg/<timestamp>.<extension>
 - Windows: <SAPDATA_HOME>\sapreorg\<timestamp>.<extension>
 See the example below. To find the extension, refer to the documentation for the procedure that you want to restart. You can also find this information in [SAPDBA Scripts and Files \[Page 214\]](#).
 - [Restart file for reorganizations \[Page 279\]](#) in the sapreorg subdirectory <timestamp>, which also contains the SQL scripts for the restart



The log 0011281141.rsi refers to reorganization of a single table or index on 28th November 2000 at 11:41. In this example on a UNIX system, you can find the restart file restart.rsi and scripts in the directory <SAPDATA_HOME>/sapreorg/012281141.

Procedure



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 despite the warnings, you are responsible for the consequences. SAPDBA continues the processing as if the error never occurred.

1. If the reorganization terminated abnormally, identify and fix the error, using the log described above in "Prerequisites". To find the log:
 - a. Choose *Show/Cleanup* → *Show log files/profiles* → *SAPDBA log files*.
 - b. Select the log <timestamp>.<extension> written during the operation that failed, and examine it to find the cause of the error.
 - c. If required, find the restart file.

For more information about the log and the restart file, see "Prerequisites" above.



Always make sure you have found the correct log and restart file before you restart a reorganization.

2. If the reorganization terminated abnormally, identify **when** the error occurred and decide how to proceed:

24.5 Reorganization with SAPDBA

When the error occurred	How to proceed
Preparation phase, that is, during the generation of the reorganization scripts	Repeat the reorganization from the beginning, but do not try to restart or reset it. Nothing has been altered in the database and a restart is not possible because not all the required scripts are present.
Initial processing of scripts before constraints have been deleted and tables renamed	Restart the reorganization.
Initial processing of scripts after constraints have been deleted and tables renamed	Restart the reorganization. Reset the reorganization if you do not want to restart it.
Later processing of scripts, when tables, indexes, or tablespaces have already been deleted	Restart the reorganization. You can not reset the reorganization.

- From the SAPDBA main menu choose *Reorganization* → *Restart*.
- Enter the *Working directory*, which is the same as you used to start the reorganization originally.
- Enter the *Restart directory*.

The default value for the *Restart subdirectory* is the `<timestamp>` directory of the last operation performed. If required, change this to the directory for the reorganization that you want to start.

SAPDBA displays a short description of the operation in the *function* field.
- To see which scripts are to be processed by the restart, select *Show restart plan*. If in doubt, we recommend you to use this option and look closely at the restart plan.
- If you want to reset the reorganization – only possible if indexes or tablespaces have **not** already been deleted – choose *Reset*.

If possible, SAPDBA resets the database to the state it was in before the failed reorganization. For example, SAPDBA sets tables that were renamed back to their original names and recreates deleted constraints. SAPDBA automatically runs the `con<TSP>.sql` and `ren<TSP>.sql` scripts during a reset.



Resetting a reorganization is usually only worthwhile when you do **not** want to restart the reorganization. If, despite this, you still want to restart after the reset, then you must manually set back all the 'Y' status indicators to 'N' in the restart plan before restarting.

- Choose *Restart* to restart the reorganization immediately or *Restart in Batch* if you want to restart it in the background or later on.

SAPDBA restarts the reorganization.

Result

On successful completion, SAPDBA displays a confirmation message.



Depending on the type of reorganization that you restarted, you might need to [back up the database \[Page 252\]](#) or at least the reorganized tablespaces. Refer to the documentation on the procedure that you are restarting.



Restart File for Reorganizations

Definition

The restart file contains the scripts required for performing the corresponding reorganization of the Oracle database, 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
```

Structure

Restart files are named as follows:

```
<SAPDATA_HOME>/sapreorg/<timestamp>/restart.<ext>
```

The extension `<ext>` indicates the SAPDBA function of the restart file:

File Extensions for a Reorganization

Extensi on	Meaning
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,
`<SAPDATA_HOME>/sapreorg/<timestamp>.<ext>`.

For a complete overview of the scripts that are written and used by SAPDBA, refer to [SAPDBA Scripts and Files \[Page 214\]](#).

See also:

- [Reorganization of a Single Object \[Page 260\]](#)
- [Reorganization of a List of Objects \[Page 263\]](#)
- [Reorganization of a Tablespace \[Page 267\]](#)
- [Reorganization of a Tablespace with Data Files \[Page 271\]](#)
- [Moving/Renaming the Data Files of a Tablespace \[Page 274\]](#)

Use

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.

Script Indicators in the Restart Files for Reorganization

Script indicat or	Meaning
-------------------------	---------

24.5 Reorganization with SAPDBA

Y	Script processed successfully
N	Script not yet processed or failed. Before a reorganization is started, all scripts have this indicator.
S	The data import has already been started but SAPDBA could not process the import script properly due to an external error (for example, import interrupted by user or power failure). SAPDBA terminated. Scripts indicated with S will automatically be newly processed in the event of a restart.

During a reorganization, all the scripts indicated with N are processed, and are changed to Y once the respective operation has been completed. The exception to this is the script `imp<TSP>.sh`: $N \rightarrow S \rightarrow Y$, in error condition $N \rightarrow S \rightarrow N$.

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 or import, SAPDBA only deletes the contents of the tables found in script `exp<TSP>_<n>.par` of the canceled import process (`imp<TSP>_<n>.sh`).



Do **not** change the sequence of scripts that appears in the restart file. This determines the flow of the corresponding reorganization.

See also:

[SAPDBA Scripts and Files \[Page 214\]](#)



Freespace Check for a Reorganization

Use

When you start a reorganization, SAPDBA for Oracle first checks whether the relevant tablespace contains enough space for the object or objects to be reorganized.

Integration

The freespace check is part of the following procedures:

- [Reorganizing a single table or index \[Page 260\]](#)
- [Reorganizing a list of tables and indexes \[Page 263\]](#)
- [Reorganizing a tablespace \[Page 267\]](#)
- [Reorganizing a tablespace and data files \[Page 271\]](#)

Prerequisites

SAPDBA takes reserve space into account when calculating the space required for the reorganized table or index. This reserve prevents the reorganization 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.

24.5 Reorganization with SAPDBA

The reserve space corresponds to the size of the `NEXT` extent of the affected table or index, but is always at least 120KB.

Features

The freespace check always allows enough space for:

- All the extents to be generated according to the number of `MINEXTENTS` for all the tables to be reorganized.
- 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 the option *ReduceOb: yes*, which means “reduce object size”).
- All expected `INITIAL` and `NEXT` extents of all **indexes** to be reorganized, calculated as in the previous two points.
- Reserve space, equal to the size of the largest `NEXT` extent or at least 120 KB.

Activities

The freespace checks are described below.



If you choose *Reorganizing a List of Tables and Indexes*, then the freespace check used by SAPDBA depends on the list:

- If the list comprises **one** object per tablespace, SAPDBA uses the freespace check “Reorganizing Tables or Indexes.”
- If the list comprises **more than one** object per tablespace, SAPDBA uses the freespace check “Reorganizing a Tablespace.”

Reorganizing Tables or Indexes

This description applies to the export/import type of reorganization. If you are using the `CREATE TABLE AS SELECT` method, see “Reorganizing Using Create Table as Select ...” below.

1. SAPDBA checks whether the extents specified by `MINEXTENTS` for all the tables to be reorganized fit into the data files in the tablespace.
2. SAPDBA also checks whether there is enough space for all the `NEXT` extents that are expected during the reorganization.
3. For the above checks, SAPDBA determines the size of the freespace areas that will be available in the individual data files when the table or index is deleted.
4. If there is not enough freespace, SAPDBA does not perform the reorganization. In this case, try either or both of the following before repeating the reorganization:
 - Extend the tablespace.
 - Use the reorganization option *Reduce object size: yes*.
5. If not enough reserve space (see “Prerequisites” above) is available, you can still start the reorganization by simply ignoring the warning message. Alternatively, you can cancel the reorganization.
6. If you are reorganizing an index or an index is reorganized as part 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.

Reorganizing a Tablespace

24.5 Reorganization with SAPDBA

This description applies to the export/import type of reorganization. If you are using the `CREATE TABLE AS SELECT` method, see “Reorganizing Using Create Table as Select ...” below.

1. 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:
 - a. Sorts the tables or indexes for reorganization in ascending `INITIAL` value order. This sequence is stored in the SQL script containing the `CREATE TABLE/INDEX` commands.
 - b. 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, but this is then gradually reduced by the size of the added objects).
 - c. Attempts to fit the extents of the tables or indexes (in the given order) in a file. This starts at the file with the most freespace. If a file with sufficient freespace is found, the table or index is added to it.
 - d. The previous two steps are repeated until space has been found in the tablespace for all objects to be reorganized.
2. If there is not enough freespace, SAPDBA does not perform the reorganization. In this case, try either or both of the following before repeating the reorganization:
 - Extend the tablespace.
 - Use the reorganization option *Reduce object size: yes*.
3. If not enough reserve space (see “Prerequisites” above) is available, you can still start the reorganization by simply ignoring the warning message. Alternatively, you can cancel the reorganization.

Reorganizing a Tablespace and Data Files

1. SAPDBA performs the checks described above in “Reorganizing a Tablespace.”
2. SAPDBA checks whether the selected file system has sufficient freespace for the new files of the tablespace. SAPDBA first suggests the file system containing file `<tablespace name>.data1`. The check considers the freespace that will be available once the tablespace is deleted.
3. SAPDBA specifies which directories the new data files for the tablespace are to 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 that can be displayed by SAPDBA. In this case, SAPDBA repeats the import check and then performs the reorganization.

Follow the [SAP Naming Conventions for Tablespaces and Data Files \[Page 49\]](#) wherever possible.



If the files are not in a standard SAP directory, SAPDBA proposes the directory `<SAPDATA_HOME>/sapdata<n>/<tablespace name>_<file number>`, which is a symbolic link to the specified directory.

4. If a raw device is to 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 change this suggestion manually.

Reorganizing Using Create Table as Select, in Deferred Mode, and During Index Rebuild

When you reorganize with `CREATE TABLE AS SELECT` or in deferred mode, SAPDBA reorganizes the objects one after the other. Sufficient disk space must be available during reorganization for both the old and the new objects. SAPDBA checks each time whether there is enough space to regenerate the largest object to be reorganized, and always reorganizes the largest object first.

In exceptional cases – for example, when the objects to be reorganized are very fragmented or free space is relatively short – Oracle errors can occur when generating the new objects during the reorganization of a list of objects or a whole tablespace. The affected object is not reorganized. You can restart the reorganization and then reorganize the affected object later on its own.



Background Reorganization

Use

You can [reorganize the Oracle database with SAPDBA \[Page 242\]](#) in the background rather than immediately.

Integration

Background reorganization is available for the following reorganization types:

- [Reorganizing a single table or index \[Page 260\]](#)
- [Reorganizing a list of tables and indexes \[Page 263\]](#)
- [Reorganizing a tablespace \[Page 267\]](#)
- [Reorganizing a tablespace and data files \[Page 271\]](#)
- [Moving or renaming data files of a tablespace \[Page 274\]](#)

Prerequisites

Make sure that you understand the UNIX `at` command. For more information, see [Configuring the UNIX Command at \[Page 186\]](#). The jobs created with `at` are transparent and can therefore be displayed using operating system commands. You can find the necessary environment variables in this kind of file, as well as the command `sapdba -r <complete path/timestamp>`, to start the reorganization.

Features

Tables and indexes that were created *before* the reorganization starts but *after* SAPDBA has created the scripts are not taken into consideration during a background reorganization. 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. Therefore, **only plan a background reorganization at a later time if no new objects were created in the meantime.**

Activities

- When scheduling several reorganizations one after the other at a later time, make sure that:
 - The export dump files for the individual reorganizations are on different disks.

24.5 Reorganization with SAPDBA

- In the case of a reorganization with data files, the old and new files of the tablespaces are on different disks.
- When scheduling several reorganizations one after the other, leave enough time between so that they do not overlap. The runtime entries in the [central log \[Page 207\]](#) `reorg<DBSID>.log` help you with scheduling.
- Do **not** perform several reorganizations in parallel, as this can lead to bottlenecks in the Oracle Dictionary or an incorrect space check.



Reorganization Methods

Use

When you [reorganize the Oracle database with SAPDBA \[Page 242\]](#), you can use either of the following methods if you are reorganizing tables:

- Export and import using Oracle commands
SAPDBA exports the data to be reorganized into an external file and then imports it back into the database.
- Data transfer using SQL statement `CREATE TABLE ... AS SELECT`
SAPDBA creates a new table in the database and transfers the contents of the old table into the new table. Finally, SAPDBA deletes the old table and renames the new one so that it has the same name as the old one.

Features

The following table summarizes the reorganization methods:

Feature	Data transfer using SQL CREATE ...	Export and import
Complexity	Low because data only processed inside database	High because data also processed outside database
Integrity	Sometimes poor because freespace of tablespace cannot be allocated optimally due to extra copy of object to be reorganized – the result can be fragmentation	Good
Scope	Cannot be used for tables with long columns and when reorganizing data files	No restrictions
Security	Good because entire operation takes place inside database	Some risk because data is exported to the file system
Space	High requirement inside database because copy of object is created	Inside database: low requirement In file system: high requirement
Speed	Fast, especially for transferring a single table to another tablespace	Slow



Storage Parameters for Reorganization and Export/Import

Definition

24.6 Export and Import of Database Objects with SAPDBA

The storage parameters below tell SAPDBA for Oracle how to perform a reorganization or an export/import.

Structure

Parameter	Default	Meaning
<i>ComprExt</i>	YES	<p>Compress all extents into one extent.</p> <p>SAPDBA calculates the size of the combined extent required from the sum of all extents currently allocated to the object.</p> <p>If this sum exceeds the maximum amount of free space available (for reorganization or export/import of a table or an index) or the maximum data file size (for reorganization or export/import of a tablespace), 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 <code>INITIAL</code> and <code>NEXT</code> are equal, and SAPDBA sets <code>MINEXTENTS</code> to the number of extents required. SAPDBA also recommends that you reduce the value for <code>NEXT</code> after the reorganization or export/import.</p> <p>The default is <i>NO</i> for Reorganizing a Tablespace [Page 267] when you use the create as select reorganization method.</p>
<i>SAP-NEXT</i>	YES	Determine the default NEXT extent size [Page 257] using the ABAP dictionary.
<i>CheckExt</i> (not for export/import)	ALL	<p>Check the freespace [Page 280] for all extents.</p> <p>The freespace check can sometimes be unnecessarily restrictive: SAPDBA states that there is not enough freespace, when there actually is. This occurs especially when <i>ComprExt: No</i> or <i>ReduceOb: No</i> are set.</p> <p>To overcome this limitation you can select <i>CheckExt: ONLY</i>. This means that only the initial extent or the number of extents specified by <code>MINEXTENTS</code> are included in the storage space check. Other next extents are not considered by the space check.</p>
<i>ReduceOb</i>	NO	<p>Reduce the object size.</p> <p>Set to <i>YES</i> if you want SAPDBA to try automatically to reduce the size allocated to the objects during the reorganization or for an export/import. To determine the actual storage space occupied, SAPDBA uses the Oracle command <code>ANALYZE TABLE <Name> ESTIMATE STATISTICS</code> or <code>ANALYZE INDEX <Name> VALIDATE STRUCTURE</code>. 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.</p>
<i>Manually</i>	NO	<p>Change the storage parameters manually [Page 254].</p> <p>Set to <i>YES</i> to change the Oracle storage parameters – such as <code>INITIAL</code>, <code>NEXT</code>, and so on – manually.</p>

24.6 Export and Import of Database Objects with SAPDBA

24.6 Export and Import of Database Objects with SAPDBA

Use

You can use SAPDBA for Oracle to export and import database objects. SAPDBA uses Oracle export and import functionality for:

- 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 database objects

You can import objects which have earlier been exported.

Export and import enables you to back up database objects in addition to other database backups. If you only want to back up particular tables, export is a good method. For example, you can add to the data backup you perform before a reorganization by exporting the objects that are to be reorganized.

This function is only intended for use with objects in a **single** database.



Do **not** use this function 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. Therefore, if you attempt to transport data between systems with this function, you end up with inconsistencies in the SAP System.

Use the SAP correction and transport system to transport objects between SAP Systems. Create new SAP databases using the SAP installation procedure.



Do **not** use this function for restore.

The data backups from an export are **logical** backups. This means that you **cannot** use them as part of an Oracle restore. The exported objects are static and are only consistent with the database if it remains unchanged.

Prerequisites

You need to know how to use SAPDBA and to have set it up correctly. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Features

There are the following [types of export \[Page 288\]](#):

- Export of tables with their indexes and data
- Export of definitions for tables and indexes
- Export of constraints, synonyms, sequences, views, and grants

You can also perform [parallel export and import \[Page 289\]](#).



SAPDBA supports export and import of partitioned tables and indexes. The export and import does **not** change the partitions and their parameters.

If a partition of a partitioned table or index is in a tablespace that you want to export and import, SAPDBA exports and imports all other partitions of the object in other tablespaces too, even if you do not specify that you want to

24.6 Export and Import of Database Objects with SAPDBA

reorganize the other tablespaces. In other words, SAPDBA exports and imports **all** partitions of a partitioned object.

Activities

What You Do

1. You [prepare \[Page 287\]](#).
2. You [perform the export \[Page 290\]](#).
3. You [perform the import \[Page 297\]](#).
4. If required, you [restart the export or import \[Page 300\]](#).

What SAPDBA Does

SAPDBA performs the export phase and, if required, the import phase too. SAPDBA generates scripts for both phases, containing the required commands. If necessary, you can start these scripts manually.

There are the following phases:

1. SAPDBA creates scripts to export and import the objects that you specify.
2. SAPDBA uses the export scripts from the first phase to export the specified objects.
3. If required, SAPDBA uses the import scripts from the first phase to import the specified objects.

24.6.1 Preparations for Export and Import

Use

This section describes the preparations necessary for an [export and import \[Page 285\]](#) with SAPDBA for Oracle.

Prerequisites

You have chosen an [export type \[Page 288\]](#).

Procedure

1. You decide if you want to perform a [parallel export and import \[Page 289\]](#).
2. You check storage space in the dump directory or 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 export is running in parallel – that is, if the value of [exp_imp_degree \[Page 201\]](#) is greater than 1 and you have specified multiple dump destinations (disk directories or tapes) – SAPDBA tries to distribute the exported objects equally across the specified directories or tapes.

3. You check that the target tablespaces exist.

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.

24.6 Export and Import of Database Objects with SAPDBA

Oracle indexes in the SAP System are generally held in tablespaces – on separate disks – other than those of the tables to which they refer.

4. You check storage space in the target tablespaces.

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.

Result

You can now start the export and import.



Types of Export and Import

Definition

The section describes the types of export and import that you can perform with SAPDBA for Oracle. For more information, see [Export and Import of Database Objects with SAPDBA \[Page 285\]](#).

Use

There are the following types of export and import:

Export and Import Type Descriptions

Type	How to Export	How to Import
Tables with indexes and data	Choose <i>Export tables and indexes including data</i> .	Choose <i>Import tables and indexes including data</i> .
Definitions for tables and indexes	Choose <i>Export structures of tables and indexes</i> .	Choose <i>Import structures</i> .
Constraints, synonyms, sequences, views, and grants	Choose <i>Export constraints, synonyms, sequences, views, grant</i> .	Choose <i>Import structures</i> .

The following table gives an overview of the individual data and application flows:

Export and Import Procedures

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 the associated table. SAPDBA also exports all constraints, comments and grants of the selected tables.</p> <p>You can optionally modify the parameters <code>INITIAL</code>, <code>NEXT</code>, <code>MINEXTENTS</code>, <code>MAXEXTENTS</code>, <code>PCTFREE</code>, <code>PCTUSED</code> and <code>FREELISTS</code> or have SAPDBA automatically calculate the value for the <code>INITIAL</code> and <code>NEXT</code> parameters (<code>PCTINCREASE</code> is always set to 0).</p> <p>SAPDBA creates import scripts that are used by SAPDBA for importing the data.</p>
Definitions for	SAPDBA creates an import script that can be used to recreate

24.6 Export and Import of Database Objects with SAPDBA

tables and indexes	<p>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 <code>INITIAL</code>, <code>NEXT</code>, <code>MINEXTENTS</code>, <code>MAXEXTENTS</code>, <code>PCTFREE</code>, <code>PCTUSED</code> and <code>FREELISTS</code> or have SAPDBA automatically calculate the value for the <code>INITIAL</code> and <code>NEXT</code> parameters (<code>PCTINCREASE</code> is always set to 0).</p>
Constraints, grants, synonyms, sequences, and views	<p>SAPDBA creates an import script that can be used to recreate these objects. The export is performed globally: all constraints, grants, synonyms, sequences, and views of the database are included (you can optionally exclude objects of users <code>SYS</code> and <code>SYSTEM</code> from this process by using <code>incl.owners SYS/SYSTEM: no</code>).</p>



Parallel Export and Import

Use

You can perform parallel export and import in SAPDBA for Oracle to speed up processing.

Prerequisites

The [exp_imp_degree \[Page 201\]](#) parameter determines the degree of parallelism for the export or import. The default value for this parameter is 1.

Parallel export and import is only relevant if you are:

- Processing **more than one** database object
- Exporting or importing **tables** and their data, **not** structures

Integration

If you are using the [export/import reorganization method \[Page 284\]](#) to reorganize more than one table or index, you can perform parallel export and import. For more information, see [Performing Processes in Parallel to Speed Up Reorganization \[Page 248\]](#).

Features

Sequential Export and Import with `exp_imp_degree = 1`

If you do not change the default value for `exp_imp_degree`, all export and import processes are executed sequentially. SAPDBA suggests the `<timestamp>` directory created in `<SAPDATA_HOME>/sapreorg` as the target for the export dump files. For more information, see the [exireo_dumpdir \[Page 200\]](#) parameter.

Parallel Export and Import with `exp_imp_degree > 1`

If several export dump files are to be created, then the export or import can be executed in parallel if you:

- Set the `exp_imp_degree` parameter to a value greater than 1. This parameter specifies the **maximum** degree of parallelism, which SAPDBA tries to achieve by using all the available directories and tape devices that you have specified.

24.6 Export and Import of Database Objects with SAPDBA

- Specify additional directories and tape devices for the export dump files. If required, you can specify the same directory several times. For more information, see *Dump destination* in [Exporting Tables and Indexes Including Data \[Page 291\]](#).

Parallel Processing

SAPDBA generates one export dump file for each parallel process, distributing 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.

SAPDBA assigns parallel processes as follows:

- If the number of parallel processes defined in export dump destination is greater than the number of tables that are to be exported, then only one process is used for each database object. The remaining directories and tape devices are not used.
- If the number of parallel processes 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 are used by the next process.

You can compress the export dump files in all the directories on disk while using parallel processing. However, the compress option has no effect when exporting to tape, for which you must use hardware compression instead.

See also:

- [SAPDBA Scripts and Files \[Page 214\]](#)
- [Logs for Parallel Export and Import Processes \[Page 210\]](#)

24.6.2 Export with SAPDBA

Use

This section tells you how to export database objects with SAPDBA for Oracle.

Prerequisites

You have completed [Preparations for Export and Import \[Page 287\]](#).



If SAPDBA uses the Oracle export program for a **tape** export, write errors are possible that the Oracle export cannot recognize. Therefore, to prevent data loss, be sure to:

- [Back up \[Page 252\]](#) the data before starting a tape export
- Use the SAPDBA option `CheckExp: yes` when exporting

In general, an export to disk causes fewer errors than one to tape.

Procedure

You perform the [type of export \[Page 288\]](#) that you identified during the preparations:

- [Exporting Tables and Indexes with Data \[Page 291\]](#)
- [Exporting Structure of Tables and Indexes \[Page 293\]](#)
- [Exporting Constraints, Synonyms, Sequences, Views, Grants \[Page 295\]](#)



Exporting Tables and Indexes Including Data

Use

You can use SAPDBA for Oracle to export one or more tables and indexes.

SAPDBA:

- Exports tables with their data, indexes, constraints, comments, and grants (but not synonyms, sequences, and views)
- Creates an import script with which the data can later be imported

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Export and Import \[Page 287\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the export in the <timestamp>.exd file. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

Procedure

1. Choose *Export/Import* → *Export tables and indexes including data*.
2. Enter the objects that you want to export by specifying one of the following names:
 - *Tablespace* – all tables or indexes in the tablespace are exported
 - *Owner* – all tables and indexes belonging to the owner are exported
 - *Table* – the table and its indexes are exported



To export Oracle system objects, choose *Owners*, then enter *a11*. In reply to the prompt *Including owners SYS and SYSTEM*, enter *y* for yes.

3. Enter the *Working directory*.

This is the working directory for the export. The default is <SAPDATA_HOME>/sapreorg or <SAPREORG>, based on the environment variable SAPREORG. Logs and scripts are saved in this directory. For more information on <SAPREORG>, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

4. Enter the *Dump destination*.

This is the directory used to store the exported data. The default is <SAPDATA_HOME>/sapreorg, specified by the [exireo_dumpdir \[Page 200\]](#) parameter (highest priority) or the SAPREORG environment variable (lower priority). For more information, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

Make sure that the space available in the directories or on the tapes is at least as large as the total size of all the objects to be reorganized. SAPDBA warns you if there is not enough space available, but you can still continue with the export if you want.

24.6 Export and Import of Database Objects with SAPDBA

If you are exporting to tape, SAPDBA asks you to state the size of the tape. SAPDBA also checks in this case whether the data fits on the tape. SAPDBA can only use one tape drive for each export dump file. For a tape export, be sure to set *CheckExp* to *YES*.



You can specify the null device `/dev/null` for testing tables and indexes on corrupted Oracle blocks. In this case, the export is performed as usual but no export dump file is created. You can also perform this check using the SAPDBA command option `sapdba -export <tablespaces/table>`. See [SAPDBA Command Mode \[Page 344\]](#).

5. Select the required parameters for *ORACLE exp/imp*:

Parameter	Default	Meaning
<i>ComprDmp</i>	<i>NO</i>	Compress the dump file, only possible for export to disk If selected, SAPDBA sends the data to the export dump files using the UNIX <code>compress</code> command. Do not use this parameter for tablespaces with objects already compressed by the SAP database interface, since it has no advantages (there might even be disadvantages).
<i>Chop</i>	<i>NO</i>	Chop the dump file, not possible for Windows platforms Select this if the export dump files are larger than the maximum file size (normally 2 GB) for your operating system. SAPDBA sends the export data to a chop tool (such as BRTOOLS), which splits the data into several smaller files.
<i>CheckExp</i>	<i>YES</i>	Check dump files, recommended especially if exporting to tape SAPDBA performs a read check after the export using the <code>inx<TSP>.sql</code> scripts.
<i>Commit</i>	<i>YES</i>	Commit command passed to database once buffer data has been imported
<i>Direct path export</i>	<i>YES</i>	Export data directly, without using SQL area This improves performance because the data is physically written straight to disk. In general, we recommend it.
<i>Parallel</i>	<i>1</i>	Export with parallel processing By increasing this to 2 or more, SAPDBA recreates the tables in parallel using the number of processes you enter. For more information, see Parallel Export and Import [Page 289] .
<i>Buffer size</i>	<i>3000000</i>	To accelerate the export, we recommend providing at least 3 MB of buffer space.

6. Enter the required [storage parameters \[Page 296\]](#) under *Storage parameters*.

7. Choose *Start* to prepare for the export.

SAPDBA prepares for the export by:

- Warning you if the SAP System is still running
- Checking for space in the dump file
- Generating the:

24.6 Export and Import of Database Objects with SAPDBA

- `<timestamp>` subdirectory in the working directory
- `<timestamp>.exd log`
- The scripts and parameter files listed below, including one to perform the import later
- `restart.exd` file for restart in the `<timestamp>` subdirectory

SAPDBA creates the following scripts and parameter files:

- `alt<TSP>.sql`
- `exp<TSP>.sh` (`exc<TSP>.sh` if necessary)
- `exp<TSP>.par` (`exc<TSP>.par` if necessary)
- `tab<TSP>.sql`
- `imp<TSP>.sh` (`imc<TSP>.sh` or `imx<TSP>.sh` if necessary) `ind<TSP>.sql` (`inx<TSP>.sql` if necessary)
- `con<TSP>.sql`
- `restart.exd` (valid both for the export and for the possible import)

8. Choose how to start the export:

- Immediately in the foreground
- In the background
- Not now, but possibly later



If you start the export in the background, tables generated after the export scripts have been created are **not** exported.

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the export \[Page 300\]](#) after fixing the problem.

If necessary, you can [import \[Page 298\]](#) the exported data. You can also [import structures singly \[Page 217\]](#) using SQL scripts.



Exporting Structures of Tables and Indexes

Use

You can use SAPDBA for Oracle to export structures of tables and indexes. Structures are the table and index definitions.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Export and Import \[Page 287\]](#).
- You have [started the database \[Page 218\]](#). The export does not affect other users and can be performed while the SAP System is running.

24.6 Export and Import of Database Objects with SAPDBA

- SAPDBA logs the export in the `<timestamp>.exs` file. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

Procedure

3. Choose *Export/Import* → *Export structures of tables and indexes*.
4. Enter the objects that you want to export by specifying one of the following names:
 - *Tablespace* – the structures of all tables or indexes in the chosen tablespace are exported
 - *Owner* – the structures of all tables and indexes belonging to the owner are exported
 - *Table* – the structures of the chosen table and its indexes are exported



To export Oracle system objects, choose *Owners*, then enter `a11`. In reply to the prompt *Including owners SYS and SYSTEM*, enter `y` for yes.

8. Enter the *Working directory*.
This is the working directory for the export. The default is `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).
9. Enter the required [storage parameters \[Page 296\]](#) under *Storage parameters*.
10. Choose *Start* to prepare for the export.
SAPDBA prepares for the export by:
 - Checking for space in the dump file
 - Generating the:
 - `<timestamp>` subdirectory in the working directory
 - `<timestamp>.exs` log
 - The scripts and parameter files listed below, including one to perform the import later
 - `restart.exs` file for restart in the `<timestamp>` subdirectory

SAPDBA creates the following scripts and parameter files:

- `tab<TSP>.sql`
 - `ind<TSP>.sql`
 - `con<TSP>.sql`
 - `grants.sql`
 - `restart.exs`
6. Choose how to start the export:
 - Immediately in the foreground
 - In the background
 - Not now, but possibly later

24.6 Export and Import of Database Objects with SAPDBA



If you start the export in the background, tables generated after the export scripts have been created are **not** exported.

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the export \[Page 300\]](#) after fixing the problem.

If necessary, you can [import \[Page 299\]](#) the exported data. You can also [import structures singly \[Page 217\]](#) using SQL scripts.



Exporting Constraints, Synonyms, Sequences, Views, and Grants

Use

You can use SAPDBA for Oracle to export the following items for the entire database:

- Constraints
Object requirements, such as interval of values allowed for a field
- Synonyms
Secondary identifications of defined database objects, such as official synonyms for views (that is, 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

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Export and Import \[Page 287\]](#).
- You have [started the database \[Page 218\]](#). The export does not affect other users and can be performed while the SAP System is running.
- SAPDBA logs the export in the <timestamp>.exg file. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#)
- All user objects – and if necessary system objects – are exported

Procedure

5. Choose *Export/Import* → *Export constraints, synonyms, sequences, views, grants*.
6. Select whether you want to include the owners `SYS` and `SYSTEM`.
11. Enter the *Working directory*.

24.6 Export and Import of Database Objects with SAPDBA

This is the working directory for the export. The default is <SAPDATA_HOME>/sapreorg or <SAPREORG>, based on the environment variable SAPREORG. Logs and scripts are saved in this directory. For more information on <SAPREORG>, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

12. Choose *Start* to start the export.

SAPDBA performs the export, creating the following scripts and parameter files:

- cssvg<DBSID>.sql
- restart.exg

Result

On successful completion, SAPDBA displays a confirmation message.

If there has been an error, SAPDBA displays the appropriate error message. You can [restart the export \[Page 300\]](#) after fixing the problem.

If necessary, you can [import \[Page 299\]](#) the exported data. You can also [import structures singly \[Page 217\]](#) using SQL scripts.



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.

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).



Storage Parameters for Reorganization and Export/Import

Definition

The storage parameters below tell SAPDBA for Oracle how to perform a reorganization or an export/import.

Structure

Parameter	Default	Meaning
ComprExt	YES	<p>Compress all extents into one extent.</p> <p>SAPDBA calculates the size of the combined extent required from the sum of all extents currently allocated to the object.</p> <p>If this sum exceeds the maximum amount of free space available (for reorganization or export/import of a table or an index) or the maximum data file size (for reorganization or export/import of a tablespace), 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, and SAPDBA sets MINEXTENTS to the number of extents required. SAPDBA also recommends that you reduce the</p>

24.6 Export and Import of Database Objects with SAPDBA

		<p>value for NEXT after the reorganization or export/import.</p> <p>The default is NO for Reorganizing a Tablespace [Page 267] when you use the create as select reorganization method.</p>
SAP-NEXT	YES	Determine the default NEXT extent size [Page 257] using the ABAP dictionary.
CheckExt (not for export/import)	ALL	<p>Check the freespace [Page 280] for all extents.</p> <p>The freespace check can sometimes be unnecessarily restrictive: SAPDBA states that there is not enough freespace, when there actually is. This occurs especially when <i>ComprExt: No</i> or <i>ReduceOb: No</i> are set.</p> <p>To overcome this limitation you can select <i>CheckExt: ONLY</i>. This means that only the initial extent or the number of extents specified by MINEXTENTS are included in the storage space check. Other next extents are not considered by the space check.</p>
ReduceOb	NO	<p>Reduce the object size.</p> <p>Set to YES if you want SAPDBA to try automatically to reduce the size allocated to the objects during the reorganization or for an export/import. To determine the actual storage space occupied, SAPDBA uses the Oracle command <code>ANALYZE TABLE <Name> ESTIMATE STATISTICS</code> or <code>ANALYZE INDEX <Name> VALIDATE STRUCTURE</code>. 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.</p>
Manually	NO	<p>Change the storage parameters manually [Page 254].</p> <p>Set to YES to change the Oracle storage parameters – such as INITIAL, NEXT, and so on – manually.</p>

24.6.3 Import with SAPDBA

Use

This section tells you how to import database objects with SAPDBA for Oracle.

Prerequisites

You have completed:

- [Preparations for Export and Import \[Page 287\]](#)
- An [export \[Page 290\]](#)

Procedure

You perform the type of import required:

- [Importing Tables and Indexes Including Data \[Page 298\]](#) if you performed [Exporting Tables and Indexes Including Data \[Page 291\]](#)
- [Importing Structures \[Page 299\]](#) if you performed one of the following:
 - [Exporting Structures of Tables and Indexes \[Page 293\]](#)

24.6 Export and Import of Database Objects with SAPDBA

- [Exporting Constraints, Synonyms, Sequences, Views, and Grants \[Page 295\]](#)



Importing Tables and Indexes Including Data

Use

You can use SAPDBA for Oracle to import one or more tables and indexes, including constraints, comments, and grants. SAPDBA created the required scripts for the import when you performed [Exporting Tables and Indexes Including Data \[Page 291\]](#). Therefore, the import uses the same settings used for the export.



You can **not** use data from an import to restore the database. This is because exported data is a **logical** backup, which is static. You can only restore the database with a proper backup. For more information, see [Database Restore and Recovery with SAPDBA \[Page 321\]](#).

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Export and Import \[Page 287\]](#).
- You have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the import in the `<timestamp>.imd` file. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

Procedure

7. Choose *Export/Import* → *Import tables and indexes including data*.

SAPDBA displays the:

- *Working directory*

This is the working directory for the import. The default is `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

- *Import subdirectory*

This is the subdirectory `<timestamp>` that contains the import scripts. This information is based on the last export performed. Compare this information with that in the appropriate export log `<timestamp>.exd`.

- *Function*

This shows the type of import to be performed, based on the type of the last export.

2. If required, change the *Working directory* or *Import subdirectory*.
3. Choose *Show import plan* to check the import plan before you start.
4. Choose *Start* to start the import, or *Start in batch* to execute the import in the background.

If there are potential problems with the import, SAPDBA warns you, so that you can stop the import.

24.6 Export and Import of Database Objects with SAPDBA

Result

SAPDBA performs the import and displays a confirmation message.

If an error occurs, SAPDBA displays an error message. You can [restart \[Page 300\]](#) the import after fixing the problem.

**Importing Structures****Use**

You can use SAPDBA for Oracle to import structures. SAPDBA creates the required scripts for the import when you perform [Exporting Structures of Tables and Indexes \[Page 293\]](#) and [Exporting Constraints, Synonyms, Sequences, Views, and Grants \[Page 295\]](#). Therefore, the import uses the settings you made for the export.

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Export and Import \[Page 287\]](#).
- You have [started the database \[Page 218\]](#). The export does not affect other users and can be performed while the SAP System is running.
- SAPDBA logs the import in the `<timestamp>.ims` file. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

Procedure

8. Choose *Export/Import* → *Import structures of database objects*.

SAPDBA displays the:

– *Working directory*

This is the working directory for the import. The default is `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

– *Import subdirectory*

This is the subdirectory `<timestamp>` that contains the import scripts. This information is based on the last export performed. Compare this information with that in the appropriate export log `<timestamp>.exs` or `<timestamp>.exg`.

– *Function*

This shows the type of import to be performed, based on the type of the last export.

2. If required, change the *Working directory* or *Import subdirectory*.
3. Choose *Show import plan* to check the import plan before you start.
4. Choose *Start* to start the import, or *Start in batch* to execute the import in the background.

If there are potential problems with the import, SAPDBA warns you, so that you can stop the import.

24.6 Export and Import of Database Objects with SAPDBA

5. Choose *Start* to start the import, or *Start in batch* to execute the import in the background.

If there are potential problems with the import, SAPDBA warns you, so that you can stop the import.

Result

SAPDBA performs the import and displays a confirmation message.

If an error occurs, SAPDBA displays an error message. You can [restart \[Page 300\]](#) the import after fixing the problem.

24.6.4 Restarting an Export or Import

Use

You can restart an export or import with SAPDBA for Oracle:

- If the export or import failed due to a problem in the export or import phase, but **not** if the failure was in the preparatory phase (that is, when scripts are created). You can restart from the point of failure.
- If you stopped the export or import after the preparatory phase because you wanted to restart it later

You can use SAPDBA for Oracle to restart any of the following procedures:

- [Exporting Tables and Indexes Including Data \[Page 291\]](#)
- [Exporting Structures of Tables and Indexes \[Page 293\]](#)
- [Exporting Constraints, Synonyms, Sequences, Views, and Grants \[Page 295\]](#)
- [Importing Tables and Indexes Including Data \[Page 298\]](#)
- [Importing Structures \[Page 299\]](#)

Prerequisites

- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You have completed the [Preparations for Export and Import \[Page 287\]](#).
- If you are exporting or importing tables and indexes including data, you have [shut down the Oracle database \[Page 219\]](#) and then [started it in restricted session mode \[Page 221\]](#).
- SAPDBA logs the import in the <timestamp>.imd file. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).
- You have identified the:
 - Log written by SAPDBA during the export or import

You can use SAPDBA to do this as described in "Procedure" below or find it yourself on disk as follows (these are default locations):

 - UNIX: <SAPDATA_HOME>/sapreorg/<timestamp>.<extension>
 - Windows: <SAPDATA_HOME>\sapreorg\<timestamp>.<extension>

24.6 Export and Import of Database Objects with SAPDBA

See the example below. To find the name of the extension, refer to the documentation for the procedure that you want to restart. You can also find this information in [SAPDBA Scripts and Files \[Page 214\]](#).

- [Restart file for export or import \[Page 302\]](#) in the `sapreorg` subdirectory `<timestamp>`, which also contains the SQL scripts for the restart



The log `0011281141.exd` refers to an export of tables and indexes including data on 28th November 2000 at 11:41. In this example on a UNIX system, you can find the restart file `restart.exd` and scripts in the directory `<SAPDATA_HOME>/sapreorg/012281141`.



Always make sure you have found the correct log and restart file before restarting an export or import.

Procedure

1. If the export or import failed, identify and fix the error, using the log described above in "Prerequisites". To do this with SAPDBA:
 - a. Choose *Show/Cleanup* → *Show log files/profiles* → *SAPDBA log files*.
 - b. Select the log `<timestamp>.<extension>` written during the operation that failed, and examine it to find the cause of the error.
 - c. If required, find the restart file `restart.<extension>`.

For more information about the log and the restart file, see "Prerequisites" above.

2. Choose *Export/Import* → *Restart*.

SAPDBA displays the:

- *Working directory*

This is the working directory for the import. The default is `<SAPDATA_HOME>/sapreorg` or `<SAPREORG>`, based on the environment variable `SAPREORG`. Logs and scripts are saved in this directory. For more information on `<SAPREORG>`, see [Environment Variables \(UNIX\) \[Page 51\]](#) and [Environment Variables \(Windows NT\) \[Page 53\]](#).

- *Restart subdirectory*

This is the subdirectory `<timestamp>` that contains the restart scripts. This information is based on the last export or import performed. Compare this information with that in the appropriate log `<timestamp>.exd`, `<timestamp>.imd`, `<timestamp>.exs`, `<timestamp>.ims`, `<timestamp>.exg`.

- *Function*

This shows the type of restart to be performed, based on the type of the most recent import or export.

9. If required, change the *Working directory* or *Restart subdirectory*.
10. Choose *Show import plan* or *Show restart plan* to check the plan before you start.
11. Choose *Restart* to restart the import or export, or *Restart in batch* to execute it in the background.

SAPDBA restarts the import or export.

24.6 Export and Import of Database Objects with SAPDBA

Result

On successful completion, SAPDBA displays a confirmation message



Restart File for Export or Import

Definition

The restart file contains the scripts that are processed during an export or an import with the Oracle database. The scripts can include the following:

```
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.
```

Structure

Restart files are named as follows:

```
<SAPDATA_HOME>/sapreorg/<timestamp>/restart.<ext>.
```

The extension `<ext>` shows the SAPDBA function that the scripts in this file were written for.

Restart File Extensions for an Export or an Import

Extension	Meaning
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 extensions are identical with those of the corresponding export log:

```
<SAPDATA_HOME>/sapreorg/<timestamp>.<ext>.
```

Logs with the following extensions are also created during the import:

Log File Extensions for an Import

Extension	Meaning
imd	Log for importing tables with indexes and data
ims	Log for structure import (table and index definitions, constraints, synonyms, sequences, views and grants)

For more information on the scripts used by SAPDBA, see [SAPDBA Scripts and Files \[Page 214\]](#).

Use

Script Indicators

When you start the respective operation (that is, an 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:

Script Indicators in the Restart Files for Export or Import

Script	Meaning
--------	---------

24.7 Database Backup with SAPDBA

Indicator	
Y	Script processed successfully
N	Script not yet processed or 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 (this only applies to a data import, and not to a structure import).
S	The data import was already started, but could not be correctly terminated due to an external error (such as a power failure, or termination by a kill command). Scripts marked with S are automatically processed again when there is a restart.

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.

All tables belonging to the script indicated with S are deleted before a restart of the Oracle import.

During an export, all necessary import scripts are created and included in the corresponding restart file (file extension `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.



Do **not** change the sequence of the scripts in the restart file. This defines the flow of the corresponding export or 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 that were already incompletely imported (since indexes also have to be deleted here) when a data import is restarted with Oracle Import. 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).

24.7 Database Backup with SAPDBA

Purpose

You can back up the data files and control files of your Oracle database with [SAPDBA \[Page 184\]](#). If you are performing an offline backup of the whole database, the online redo log files are also backed up. If you have lost data and want to recover it, these files are essential. You can also [back up non-database files and directories \[Page 361\]](#).

You typically use SAPDBA for a one-off backup, such as after a structure change to the database. For example, after [moving or renaming the data files of a tablespace \[Page 274\]](#), you ought to back up the database.



For routine backups, we recommend one of the following:

24.7 Database Backup with SAPDBA

- DBA Planning Calendar to schedule a backup and then view its log
- A scheduler offered by a supplier of the BACKINT interface
- The scheduler `cron` for UNIX or `at` for Windows

Like SAPDBA, these tools call the SAP tool [BRBACKUP \[Page 360\]](#) to perform the backup.

Integration

SAPDBA normally calls the SAP tool BRBACKUP. You can also perform a backup directly by calling BRBACKUP from the command line.



We recommend you to normally use SAPDBA rather than BRBACKUP. This is because the SAPDBA menus simplify entry of the correct parameters.

Features

You can perform the following backup functions with SAPDBA:

- Perform normal backup
- Initialize BRBACKUP tape
- Show BRBACKUP tape header information
- Determine compression rates
- Prepare for RMAN run
- Verify BRBACKUP tape

For the function Verify BRBACKUP tape, SAPDBA calls [BRRESTORE \[Page 395\]](#).

Activities

1. You call the backup function in SAPDBA and check the displayed backup parameters, changing them as required.

The default values for the backup parameters, which are set in the [initialization profile `init<DBSID>.sap` \[Page 445\]](#), trigger an **offline full** database backup to a **local tape device without file compression**. This means that the online redo log files and control file are backed up as well as the data files.



SAPDBA only lets you change certain parameters for the backup. If you have to make other changes, you must change the `init<DBSID>.sap` profile manually and then restart SAPDBA.

2. If required, you change the default values for the backup parameters in the initialization profile `init<DBSID>.sap` and restart SAPDBA.

3. You start the backup.

If the backup is being made locally or remotely to tapes or disks, then the backup is monitored and an estimation is made of the backup time, based on the elapsed time and the size of the files that still have to be backed up. You also see success or error messages.

4. You check the results of the backup in the [BRBACKUP logs \[Page 377\]](#).

For more information, see [Backing Up the Database with SAPDBA \[Page 304\]](#).

24.7.1



Backing Up the Database with SAPDBA

Use

You can use SAPDBA for Oracle to back up the database. For more information, see [Database Backup with SAPDBA \[Page 303\]](#).



Remember that the SAP System is not available for production work during an offline backup, since the database is closed. However, you can perform an online backup when the database system and SAP System are running.

Prerequisites

- Make sure the database is running in ARCHIVELOG mode. For more information, see [Setting Archive and Database Mode with SAPDBA \[Page 41\]](#).
- Make sure you have set the necessary BRBACKUP parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#), because SAPDBA uses these when it calls 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. For more information, see [Volume Management \[Page 91\]](#).
- Allow enough time for the backup. [Hardware compression \[Page 101\]](#) can halve the backup time.
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Choose *Backup database*.
2. Choose *Backup function* and choose the required function:

Backup Function	Meaning
<i>Normal backup</i> (default)	Performs a normal backup With this function, BRBACKUP does not initialize the tape volume, so you can start the backup immediately.
<i>Initialize BRBACKUP tape</i>	Initializes a tape volume
<i>Show BRBACKUP tape header information</i>	Displays the information in the volume label
<i>Determine compression rates</i>	Determines the compression rates of all data files specified with the parameter <i>Objects for backup</i> You need to allow some time for this procedure. If you back up to tape devices with hardware compression, we recommend you to use this option about once a month. This is so that the current compression rate of the files can be determined. For more information, see compress [Page 454] .
<i>RMAN preparation run</i>	Performs a preparation run for a backup using Oracle Recovery Manager (RMAN) [Page 137] with the SAP backup library For more information, see RMAN Save-Set Grouping [Page 151] and RMAN Tape Layout [Page 149] .
<i>Verify BRBACKUP tape</i>	Checks backup to tape

24.7 Database Backup with SAPDBA



If you choose a different function, the *Backup database* menu looks different when you return to it.

3. Depending on the function you have chosen, [set the required function-specific parameters \[Page 306\]](#), including the [special options \[Page 311\]](#).
4. To start the backup with the selected parameters, choose *Start BRBACKUP*. To perform function *Verify BRBACKUP tape*, choose *Start VERIFICATION*.
5. You check the results of the backup in the [BRBACKUP logs \[Page 377\]](#).
 - The summary log `back<DBSID>.log` displays the return code for the backup.
 - The detail log `b<encoded timestamp>.<ext>` displays the progress of the backup.

For more information on how to view the logs with SAPDBA, see [Show/Cleanup to Manage Logs and Profiles with SAPDBA \[Page 197\]](#).



Setting Parameters for SAPDBA Database Backup

Use

This section tells you how to set parameters when you are [backing up the Oracle database with SAPDBA \[Page 304\]](#).

Procedure

Set the required parameters for the chosen function (the parameters are listed in alphabetical order):

- *Backup device type*

Sets the device type used for the functions *Normal backup*, *Initialize BRBACKUP tape*, and *Show BRBACKUP tape header information*. Choose from the following options:

Option	Meaning	Corresponds to backup_dev type [Page 447] setting
<i>local tape</i> (default)	Backup to local tape device	tape
<i>local tape auto changer</i>	Backup to tape when using a device with automatic tape changer. Messages telling you to change the tape are suppressed.	tape_auto
<i>local tape juke box</i>	Backup to jukeboxes or autoloaders that can be addressed locally	tape_box
<i>remote tape</i>	Backup to tape device in a remote system, using the command specified in the initialization profile	pipe
<i>remote tape auto changer</i>	Backup to device with automatic tape changer using a remote system Messages telling you to change the tape are suppressed.	pipe_auto
<i>remote tape juke box</i>	Backup to jukeboxes or autoloaders that can be addressed remotely	pipe_box

24.7 Database Backup with SAPDBA

<i>external backup tool (backint)</i>	Backup using an external backup program [Page 180] addressed by BACKINT interface program	util_file
<i>external backup tool (backint) online</i>	File-by-file online backup using an external backup program addressed by BACKINT	util_file_online
<i>RMAN with external backup tool (backint)</i>	Backup with an external backup library and RMAN, using BACKINT For more information, see RMAN Backup with an External Backup Library [Page 144] .	rman_util
<i>RMAN with local disk</i>	Backup to local disk with an external backup library and RMAN, but without BACKINT	rman_disk
<i>RMAN with remote disk</i>	Backup to remote disk with an external backup library and RMAN, but without BACKINT	rman_stage
<i>local disk</i>	Backup to a local disk	disk
<i>local disk (create database copy)</i>	Backup with database copy to a local disk This option creates a copy of the database files, including the directory structure, for example when setting up a test system. See Structure-Retaining Database Copy [Page 111] .	disk_copy
<i>local disk (create standby database)</i>	Backup with database copy to a local disk for a standby database	disk_standby
<i>remote disk</i>	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.	stage
<i>remote disk (create database copy)</i>	Backup with database copy to a remote disk	stage_copy
<i>remote disk (create standby database)</i>	Backup with database copy to a remote disk for a standby database	stage_standby

- *Backup from disk backup*
Copies a disk backup to tape as part of a [two-phase backup \[Page 109\]](#) for function *Normal backup*. The default is *No*. Choose one of the disk backups displayed by SAPDBA.
- *Backup name*
Specifies the backup name for function *Verify BRBACKUP tape*.
- *Backup type*

24.7 Database Backup with SAPDBA

Specifies the backup type for the functions *Normal backup*, *Determine compression rates*, and *RMAN preparation run*. Choose from the following options:

Option	Meaning
<i>online</i> (default)	Backup with the database still online <i>ARCHIVELOG</i> mode must be on.
<i>offline</i>	Backup with the database offline
<i>online (consistent)</i>	Logically consistent online backup The offline redo log files generated during the backup are also copied to the same volume.
<i>offline (force)</i>	Offline backup after the database is forcibly shut down The system does not check whether SAP System users are active.
<i>offline (standby)</i>	Offline backup of standby database See Standby Database: BRBACKUP Backup of Database Files [Page 167] .
<i>online (split)</i>	Backup on backup host after mirrors are split The mirror disks are split and backed up without downtime for the production database. The production tablespaces to be backed up are only placed in <i>BACKUP</i> status during the actual split. See Split-Mirror Disk Backup [Page 170] .
<i>offline (split)</i>	Offline backup after mirrors are split The mirror disks are split and backed up with minimal downtime for the production database. The production database is only shut down for a short time for the actual split. See Split-Mirror Disk Backup [Page 173] .
<i>offline (stop)</i>	Offline backup, but without subsequent database restart

- *Backup volume(s)*

Specifies the IDs of the tape volumes that you want to use for functions *Normal backup* and *Initialize BRBACKUP tape*. If you specify more than one volume name, separate the names using commas.

To disable tape management, enter `scratch`.

- *Check volume label*

Checks the volume label for function *Initialize BRBACKUP tape*. Choose from the following:

- Yes (default)

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.

- No

The label of the volume is not checked. You can use this to initialize new non-SAP volumes or make a new label for previously initialized volumes.



The expiration period is not checked when you choose *No*.

- *Make part. backups compl.*

24.7 Database Backup with SAPDBA

Makes partial backups complete for function *Normal backup*. Use this if you have performed partial backups recently and want to back up all remaining data in the database. SAPDBA prompts you for the number of days over which you want to complete the backup.

For more information, see [-f|-fillup \[Page 369\]](#).



Your database consists of 12 tablespaces. You backed up 3 tablespaces each on Monday, Wednesday, and Friday. To back up the remaining 3 tablespaces on Saturday, set this parameter to 6.

- *Number of tapes*

Specifies the number of tapes for functions *Initialize BRBACKUP tape* and *Show BRBACKUP tape header information*.

- *Objects for backup*

Specifies the objects to be processed for functions *Normal backup* and *Determine compression rates*. Choose from the following options:

Option	Meaning
<i>all</i> (default)	Whole database backup
<i>all_data</i>	Backup of all tablespaces except pure index tablespaces
<i>full</i>	Full database backup at level 0
<i>incr</i>	Incremental database backup at level 1
<i>sap_dir</i>	Backs up the SAP directories
<i>ora_dir</i>	Backs up the Oracle non-database directories
<i><tablespace name></i>	Backs up the specified tablespace
<i><file ID></i>	Backs up one or more files that are specified with an Oracle file ID or a range of file IDs
<i><absolute file or directory name></i>	Backs up a file or directory that is specified by the absolute name
<i><combination></i>	Backs up a combination of objects that you specify by separating the individual objects with commas

For more information, see [m|-mode \[Page 370\]](#) and [Database Backup Types \[Page 80\]](#).

- *Objects for preparation run*

Specifies the objects for function *RMAN preparation run*. For more information, see parameter *Objects for Backup* above.

- *Objects for verification*

Specifies the objects for function *Verify BRBACKUP tape*. For more information, see parameter *Objects for Backup* above.

- *Parameter file*

Specifies the parameter file for the backup. The default is [init<DBSID>.sap \[Page 445\]](#). The values specified in the parameter file are used when you first choose the *Backup database* menu. Thereafter, you can change some parameters, but not all. Changes are only valid until you leave the *Backup database* menu.

To use a different initialization profile for BRBACKUP, specify the name of the profile you want to use and restart SAPDBA.

24.7 Database Backup with SAPDBA

- *Query only*

Tells you which tapes are required for the backup for function *Normal backup*. Choose from the following:

- *No* (default)

Backup starts with no query

- *Yes (query only with tape label check)*

BRBACKUP tells you which tapes to mount for the backup and checks whether you have actually mounted these tapes. There is no backup.

- *Yes (query only without tape label check)*

BRBACKUP tells you which tapes to mount for the backup but does not check whether you have actually mounted these tapes. There is no backup.

For more information, see [-q|-query \[Page 373\]](#).

- *Restart backup*

Restarts a backup that has previously failed, but from the point of failure, for function *Normal backup*. That is, data that was already backed up before the failure is **not** backed up again after the restart. Choose one of the backups displayed. The default is the latest backup.

For more information, see [-f|-fillup \[Page 369\]](#).

- *Special options*

For more information, see [Setting Special Options for SAPDBA Database Backup \[Page 311\]](#).

- *Standard backup*

Specifies whether a standard backup occurs for function *Normal backup*.

- *Yes* (default)

A standard backup means that you have **not** specified any of the following parameters:

- *Backup from disk backup*
- *Restart backup*
- *Make part. backups compl.*

- *No*

This means that you have specified one of the above parameters, in order to perform a non-standard backup.

- *Verification type*

Specifies the verification type for function *Verify BRBACKUP tape*. Choose from the following:

- *Reading tape(s)* (default)

Only the readability of the tapes is checked, not the contents. SAPDBA calls BRRESTORE for this. It reads the files but does not restore them.

For more information, see [-w|-verify \[Page 405\]](#).

- *Restore to stage and DB VERIFY*

The backup is restored to the directory [compress_dir \[Page 455\]](#). The block structure is checked with the Oracle tool DBVERIFY using `brrestore -w use_dbv`.

For more information, see [-w/-verify \[Page 405\]](#), [Backup Verify \[Page 118\]](#) and [Verify of RMAN Backups \[Page 150\]](#).



Setting Special Options for SAPDBA Database Backup

Use

This section tells you how to set special options when you are [backing up the Oracle database with SAPDBA \[Page 304\]](#).

Procedure

Set the required special options for the chosen function (the parameters are listed in alphabetical order):

- *Append archive backup*
Specifies whether the offline redo log files are backed up after the database. Set this option as follows:
 - *no* (default)
The database is backed up using BRBACKUP as usual.
 - *yes*
After the database has been backed up using BRBACKUP, BRBACKUP calls BRARCHIVE to back up the offline redo log files to the **same** backup tape.
When you choose *yes* for this option, *Configure archive backup* (see below) appears in the menu.
See also [BRBACKUP and BRARCHIVE Backups in One Run \[Page 116\]](#).
- *Check label before initialize*
Checks the volume label before the function *Initialize BRBACKUP tape*.
- *Compress*
Specifies whether [hardware \[Page 101\]](#) or [software \[Page 102\]](#) compression is used. Set this option as follows:
 - *no* (default)
No compression
 - *yes*
Software compression.
 - *hardware*
Hardware compression. First make sure your tape devices support this.
- *Configure archive backup*
Specifies parameters for backup of the offline redo log files when you have chosen *Append archive backup* (see above). For more information, see [Backing Up the Offline Redo Log Files with SAPDBA \[Page 314\]](#). Any options that are already set for the backup are hidden, including *Archive device type*, *Archive volume(s)*, and so on.
- *Confirm backup parameters*

24.7 Database Backup with SAPDBA

Specifies whether BRBACKUP confirmation messages are suppressed. Set this option as follows:

- *no* (default)
Some messages are suppressed. For more information, see [-cl-confirm \[Page 366\]](#).
- *yes*
All messages are displayed.

- *Enter password interactively*

Specifies whether you can enter the password interactively for the backup. Set this option as follows:

- *no* (default)
You cannot enter the password interactively.
- *yes*
You must enter the password interactively. The password is not visible when you enter it and does not appear in the process list.

- *Language*

Specifies the language for BRBACKUP messages. Set this option as follows:

- *English* (default)
- *deutsch* for German

- *Level of parallel execution*

Specifies the maximum number of parallel copy processes

The default is 0 and means that the number of parallel processes matches the number of backup devices.

- *Members per RMAN saveset*

Specifies the number of data files in each save set for backups with the Oracle Recovery Manager (RMAN). For more information, see [RMAN Save-Set Grouping \[Page 151\]](#).

- *Show installed versions*

Shows details of which versions of the BR tools are installed

- *Verification after backup*

Specifies whether the tapes are verified after backup. Set this option as follows:

- *no backup verification* (default)
No verification
- *binary (offline) / by size (online)*
Verifies the backup either byte by byte (for *binary*) or using the overall size of the backup (for *by size*).
- *Verify backup with DBVERIFY*
Verifies the backup using DBVERIFY. For more information, see [-w|-verify \[Page 376\]](#) with the `use_dbv` option.

24.8 Backup of the Offline Redo Log Files with SAPDBA

Use

You can back up the offline redo log files with [SAPDBA \[Page 184\]](#) for Oracle. For more information about backup, see [Backup, Restore, and Recovery \[Page 73\]](#). Oracle copies the online redo log files to the archiving directory, so creating the offline redo log files. You typically use SAPDBA for a one-off backup of the offline redo log files.



For routine backups, we recommend one of the following:

- DBA Planning Calendar to schedule a backup and then view its log
- A scheduler offered by a supplier of the BACKINT interface
- The scheduler `cron` for UNIX or `at` for Windows

Like SAPDBA, these tools call the SAP tool [BRARCHIVE \[Page 383\]](#) to perform the backup.



If you need to recover your database you must have access to all offline redo log files that have been written since the database backup. Otherwise you can only recover the database to the point in time of the last available redo log. Therefore, you must always archive the offline redo log files after a database backup, and **immediately** after an online backup.

In production systems, we strongly recommend you to:

- Run the database in ARCHIVELOG mode with automatic archival turned on. For more information, see [Setting Up Archiving \[Page 43\]](#).
- Make two copies of the offline redo log files. In test systems, one copy is often sufficient.

Integration

The Oracle database hangs if it is operated in ARCHIVELOG mode, but the archiving process cannot save the online redo log files because the archiving directory is full. This situation is called “archiver stuck.” If this occurs you must back up the offline redo log files and delete them from the archiving directory as soon as possible.

To avoid archiver stuck, back up the offline redo log files regularly to tape. How often you do this depends on the amount of activity in your SAP System. If a lot of redo log entries are written, and the redo log files are frequently switched, be sure to constantly monitor the archiving directory. When necessary, archive and delete the offline redo log files.

For more information about avoiding “archiver stuck,” see SAP Note 316642.

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 saves the online redo log files automatically as offline redo log files. Unless you have changed the standard profile `init<DBSID>.ora`, the offline redo log files are stored in the archive directory `<SAPDATA_HOME>/saparch`. For more information, see [Setting Up Archiving \[Page 43\]](#).
- You can also directly archive the offline redo log files by calling the correct [command option for BRARCHIVE \[Page 384\]](#).

24.8 Backup of the Offline Redo Log Files with SAPDBA

- Make sure the necessary parameters have been set for BRARCHIVE in the [Initialization Profile init<DBSID>.sap \[Page 445\]](#).

Activities

1. You call the function to back up offline redo logs in SAPDBA and check the displayed backup parameters, changing them as required.

The default values for the backup parameters, which are set in the [initialization profile init<DBSID>.sap \[Page 445\]](#), trigger a simple archive of all offline redo log files to a local tape device without file compression.



SAPDBA only lets you change certain parameters for the backup. If you have to make other changes, you must change the `init<DBSID>.sap` profile manually and then restart SAPDBA.

2. If required, you change the default values for the backup parameters in the initialization profile `init<DBSID>.sap` and restart SAPDBA.
3. You start the backup of the offline redo log files.
4. You check the results of the backup in the [BRARCHIVE logs \[Page 392\] \[Page 392\]](#):
 - To check the archive return code, see the summary log, `arch<DBSID>.log`.
 - To check the progress of the backup, see the detail log, `a<encoded timestamp>.<ext>`.

For more information, see [Backing Up the Offline Redo Log Files with SAPDBA \[Page 314\]](#).

24.8.1 Backing Up the Offline Redo Log Files with SAPDBA

Use

You can use SAPDBA for Oracle to back up the offline redo log files. For more information, see [Backup of the Offline Redo Log Files with SAPDBA \[Page 313\]](#).

Prerequisites

- Make sure the database is running in ARCHIVELOG mode. For more information, see [Setting Archive and Database Mode with SAPDBA \[Page 41\]](#).
- Make sure you have set the necessary BRARCHIVE parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#), because SAPDBA uses these when it calls BRARCHIVE.
- 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. For more information, see [Volume Management \[Page 91\]](#).
- Allow enough time for the backup. [Hardware compression \[Page 101\]](#) can halve the backup time.
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Choose *Backup Offline Redo Logs*.

24.8 Backup of the Offline Redo Log Files with SAPDBA

2. Choose Archive function and choose the required function:

Archive Function	Meaning
Save (default)	Performs a normal backup of any offline redo log files that have not yet been backed up
Second copy	Performs a normal backup with a second copy
Double save (parallel)	Performs a normal backup with a second copy to two tape devices in parallel
Save and delete	Performs a normal backup and then deletes the offline redo log files that have been backed up
Second copy and delete	Performs a normal backup with a second copy and then deletes the offline redo log files that have been backed up.
Double save and delete	Performs a normal backup with a second copy to two tape devices in parallel and then deletes the offline redo log files that have been backed up
Delete saved	Deletes the offline redo log files that have already been backed up (the oldest backed-up offline redo log files are deleted first).
Delete copied	Deletes offline redo log files that have already been backed up twice
Copy and save	Creates a second copy of offline redo log files that have already been backed up and then backs up the offline redo log files that have been created in the meantime
Copy, delete and save	Creates a second copy of offline redo log files that have already been backed up and then deletes them and backs up the offline redo log files that have been created in the meantime
Initialize BRARCHIVE tape	Initializes a tape volume
Show BRARCHIVE header information	Displays the information in the volume label
Stop BRARCHIVE run with fill option	Stops a BRARCHIVE backup run that was started with the option <i>Continue until end of tape safely</i> .



If you choose a different function, the *Backup Offline Redo Logs* menu looks different when you return to it.

- Depending on the function you have chosen, set the required function-specific parameters, including the special options.
- To start the backup with the selected parameters, choose *Start BRARCHIVE*.
- You check the results of the backup in the [BRARCHIVE logs \[Page 392\]](#).
 - The summary log `arch<DBSID>.log` displays the return code for the backup.
 - The detail log `a<encoded timestamp>.<ext>` displays the progress of the backup.

For more information on how to view the logs with SAPDBA, see [Show/Cleanup to Manage Logs and Profiles with SAPDBA \[Page 197\]](#).



Setting Parameters for SAPDBA Backup of Offline Redo Logs

24.8 Backup of the Offline Redo Log Files with SAPDBA

Use

This section tells you how to set parameters when you are [backing up the offline redo logs of the Oracle database with SAPDBA \[Page 314\]](#).

Procedure

Set the required parameters for the chosen function (the parameters are listed in alphabetical order):

- *Archive device type*

Sets the device type used. Choose from the following options:

Option	Meaning	Corresponds to backup dev type [Page 447] setting
<i>local tape</i> (default)	Backup to local tape device	tape
<i>local tape auto changer</i>	Backup to tape when using a local device with automatic tape changer. Messages telling you to change the tape are suppressed.	tape_auto
<i>local tape juke box</i>	Backup to local jukeboxes or autoloaders, in which tapes that can be addressed directly	tape_box
<i>remote tape</i>	Backup to tape device in a remote system, using the command specified in the initialization profile	pipe
<i>remote tape auto changer</i>	Backup to device with automatic tape changer using a remote system Messages telling you to change the tape are suppressed.	pipe_auto
<i>remote tape juke box</i>	Backup to remote jukeboxes or autoloaders, in which tapes can be addressed remotely	pipe_box
<i>external backup tool (backint)</i>	Backup using an external backup program [Page 180] addressed by BACKINT interface program	util_file
<i>external backup tool (backint) online</i>	File-by-file online backup using an external backup program addressed by BACKINT	util_file_online
<i>external backup tool with rman (backint)</i>	Backup with an external backup library and RMAN, using BACKINT to save control file, profiles, and log files For more information, see RMAN Backup with an External Backup Library [Page 144] .	rman_util
<i>external backup tool with rman (local disk)</i>	Backup with an external backup library and RMAN, where control file, profiles, and log files are saved	rman_disk
<i>external backup tool with</i>	Backup to remote disk with an external backup library and RMAN, where control	rman_stage

24.8 Backup of the Offline Redo Log Files with SAPDBA

<i>rman (remote disk)</i>	file, profiles, and log files are saved	
<i>local disk</i>	Backup to a local disk	disk
<i>remote disk</i>	Backup to a remote disk This option is recommended if no disk is installed locally, since the transfer protocol is secure (<code>ftp</code> or <code>rcp</code>), and no verification is required, unlike an NSF mount.	stage

- *Archive volume(s)*

Specifies the IDs of the tape volumes for the backup. If you specify more than one volume name, separate the names using commas.

To disable tape management, enter `scratch`.

- *Backup from disk backup*

Copies a disk backup to tape as part of a [two-phase backup \[Page 109\]](#). The default is *No*.

- *Check volume label*

Checks the volume label for function *Initialize BRARCHIVE tape*. Choose from the following:

- Yes (default)

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.

- No

The label of the volume is not checked. You can use this to initialize new non-SAP volumes or create a new label for previously initialized volumes.



The expiration period is not checked when you choose *No*.

- *Fill tape(s) permanently*

Specifies whether the offline redo log files are permanently archived, either individually or in groups. Choose from the following:

- No

An offline redo log file is archived individually as soon as it is created. This process is not stopped until the volume is full or the number of redo log files specified by *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.

This option is useful if you are using BACKINT with an external backup program.

For more information, see [Grouping Offline Redo Log Files \[Page 320\]](#) and [-f|-fill \[Page 387\]](#).

- *Number of redo logs*

Enter the number of offline redo log files that you want to process. The default is 10000.

24.8 Backup of the Offline Redo Log Files with SAPDBA



If set to 2, BRARCHIVE processes the two oldest offline redo log files in the archiving directory that have not yet been backed up to tape.
If set to 10000, BRARCHIVE normally processes all offline redo log files in the archiving directory.

- *Number of tapes*

Specifies the number of tapes for functions *Initialize BRARCHIVE tape* and *Show BRARCHIVE tape header information*.

- *Parameter file*

Specifies the parameter file for the backup. The default is [init<DBSID>.sap \[Page 445\]](#). The values specified in the parameter file are used when you first choose the *Backup Offline Redo Logs* menu. Thereafter, you can change some parameters, but not all. Changes are only valid until you leave the menu.

To use a different initialization profile for BRARCHIVE, specify the name of the profile you want to use.

- *Query only*

Tells you which tapes are required. Choose from the following:

- *No (default)*

Backup starts with no query

- *Yes (query only with tape label check)*

SAPDBA tells you which tapes to mount for the backup and checks whether you have actually mounted these tapes. There is no backup.

- *Yes (query only without tape label check)*

SAPDBA tells you which tapes to mount for the backup but does not check whether you have actually mounted these tapes. There is no backup.

For more information, see [-q|-query \[Page 389\]](#).

- *Special options*

For more information, see [Setting Special Options for SAPDBA Backup of Offline Redo Logs \[Page 318\]](#).

- *Standard backup of offline redo logs*

Specifies whether a standard backup occurs.

- *Yes (default)*

A standard backup means that you have **not** specified *Backup from disk backup*.

- *No*

This means that you have specified *Backup from disk backup*.



Setting Special Options for SAPDBA Backup of Offline Redo Logs

Use

This section tells you how to set special options when you are [backing up the offline redo logs of the Oracle database with SAPDBA \[Page 314\]](#).

24.8 Backup of the Offline Redo Log Files with SAPDBA

Procedure

Set the required special options for the chosen function (the parameters are listed below in alphabetical order):

- *Append database backup*

Specifies whether the database is backed up after the offline redo logs. Set this option as follows:

- *no* (default)

The offline redo logs are backed up using BRARCHIVE as usual.

- *yes*

After the backup of the offline redo logs, BRARCHIVE calls BRBACKUP to back up the database to the **same** backup tape.

When you choose *yes* for this option, *Configure database backup* (see below) appears in the menu.

See also [BRBACKUP and BRARCHIVE Backups in One Run \[Page 116\]](#).



We strongly recommend you to normally perform the database backup **before** the offline redo log backup, because this ensures that online database backups are logically consistent.

- *Apply logs into standby DB*

The following entries are possible:

- *no* (default)

If you do not have a standby database, or if you do not want to import the redo log files into the standby database.

- *<n>*

Specifies the number of minutes that BRARCHIVE waits before it imports the redo log files into the standby database. For more information, see [-m\]-modify \[Page 389\]](#).

- *Check label before initialize*

Checks the volume label before the function *Initialize BRARCHIVE tape*.

- *Compress*

Specifies whether [hardware \[Page 101\]](#) or [software \[Page 102\]](#) compression is used. Set this option as follows:

- *no* (default)

No compression

- *yes*

Software compression

- *hardware*

Hardware compression. First make sure your tape devices support this.

- *Configure database backup*

Specifies parameters for database backup when you have chosen *Append database backup* (see above). For more information, see [Backing Up the Database with SAPDBA](#)

24.8 Backup of the Offline Redo Log Files with SAPDBA

[\[Page 304\]](#). Any options that are already set for the backup, such as *Backup device type* and *Backup volume(s)*, are hidden.

- *Confirm backup parameters*

Specifies whether BRARCHIVE confirmation messages are suppressed. Set this option as follows:

- *no* (default)
Some messages are suppressed. For more information, see [-c|-confirm \[Page 386\]](#).
- *yes*
All messages are displayed.

- *Enter password interactively*

Specifies whether you can enter the password interactively for the backup. Set this option as follows:

- *no* (default)
You cannot enter the password interactively.
- *yes*
You must enter the password interactively. The password is not visible when you enter it and does not appear in the process list.

- *Language*

Specifies the language for BRARCHIVE messages. Set this option as follows:

- *English* (default)
- *deutsch* for German

- *Show installed versions*

Shows which versions of the BR tools are installed

- *Verification after backup*

Specifies whether the tapes are verified after backup. Set this option as follows:

- *no* (default)
No verification
- *yes*
Verification. Use this function to check the readability and the completeness of the backup. For more information, see [-w|-verify \[Page 391\]](#).

24.8.2 Grouping Offline Redo Log Files

Use

When you [back up offline redo log files \[Page 314\]](#) for your Oracle database, you can group the files. If you use an [external backup programs \[Page 180\]](#) with the BACKINT interface, we recommend you to use this function. In this case, every archiving action activates the interface and often repositions the tape, which 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 speeds up the backup.

24.9 Database Restore and Recovery with SAPDBA

By using permanent backup of the offline redo log files with `brarchive -f fill` [Page 387], you can prevent a possible overflow of the `saparch` archiving directory. You can also collect a certain number of offline redo log files before BRARCHIVE backs them all up together on tape, by using the command `brarchive -f <number>`, as shown in the following example.

Example

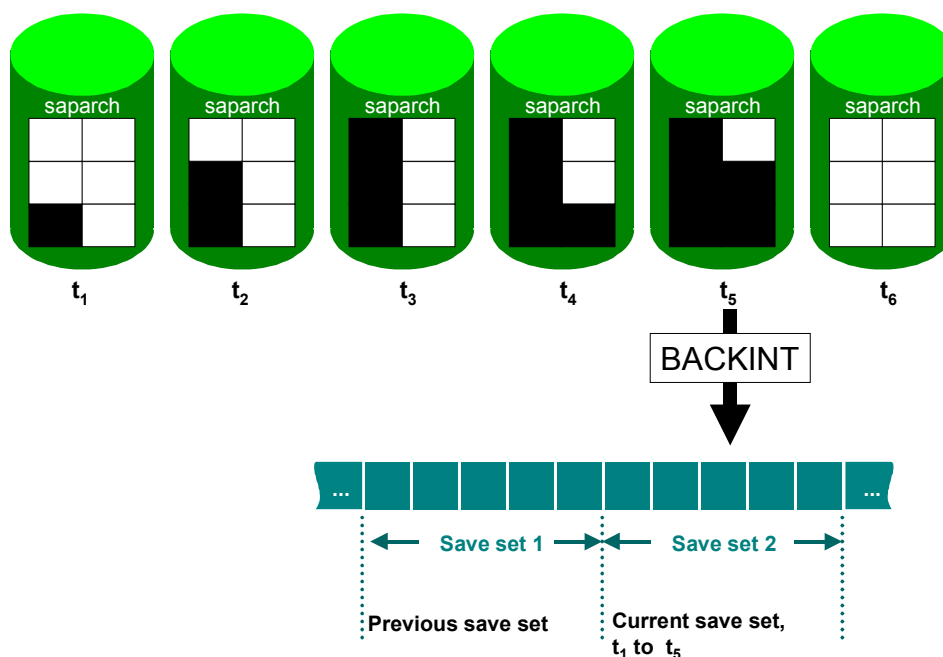


This example uses the following command:

```
brarchive -sd -d util_file -f 5
```

After five offline redo log files have been created, BRARCHIVE saves these to tape. It repeats this until either of the following is true:

- BRARCHIVE was stopped with call `brarchive -f stop`
- the maximum number of redo logs specified has been reached



24.9 Database Restore and Recovery with SAPDBA

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

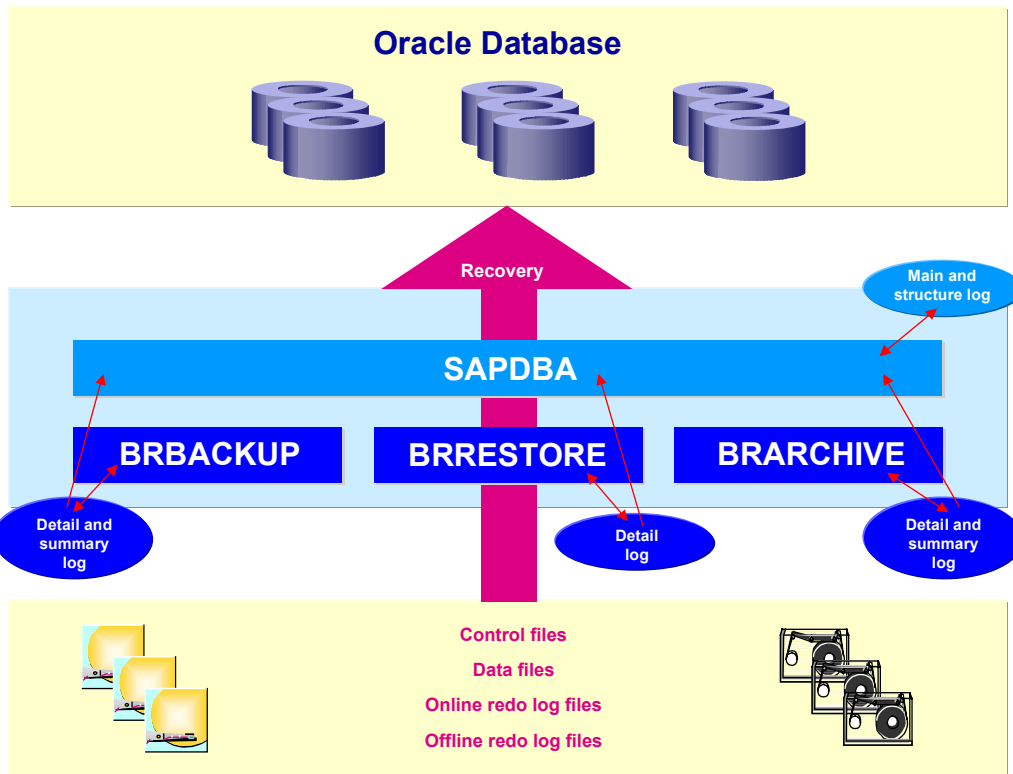
The SAPDBA restore and recovery functions are based on the backup tools BRBACKUP, BRARCHIVE, and BRRESTORE. In a recovery, the backups created with BRBACKUP and BRARCHIVE are restored with BRRESTORE. The operating system logs of BRBACKUP and BRARCHIVE are very important for the recovery, because they contain all the information

24.9 Database Restore and Recovery with SAPDBA

about the backups, such as directories, volumes, and timestamps. You might need to restore these logs from existing backups.



SAPDBA recovery is not based on BRBACKUP or BRARCHIVE logs in database tables, because these tables are not available when you recover the database.



You can recover the database from the SAPDBA *Restore/Recovery* menu.



For more information on the concepts behind restore and recovery, see [Approach to Restore and Recovery \[Page 121\]](#).

Prerequisites

For a successful recovery of the database, you **must** check that:

- The BRBACKUP and BRARCHIVE logs are 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 are available.



You might also need to be meet other prerequisites, depending on the context of the recovery.

Features

SAPDBA offers the following for recovering the database in particular situations:

- [Partial Restore and Complete Recovery with SAPDBA \[Page 323\]](#)

24.9 Database Restore and Recovery with SAPDBA

- [Performing a Full Restore and Point-in-Time Recovery with SAPDBA \[Page 335\]](#)
- [Reset Database \[Page 340\]](#)
- [Restore One Tablespace \[Page 341\]](#)
- [Restore Individual File\(s\) \[Page 341\]](#)



Familiarize yourself with the differences between these functions before you start the recovery of the database. If you choose the wrong function, you might not be able to recover the database, or you might increase downtime.

24.9.1 Partial Restore and Complete Recovery with SAPDBA

Purpose

This section tells you how to restore and recover your Oracle database with SAPDBA. This process normally corrects media errors, such as a disk crash. When such an error occurs, it usually involves the loss of database files, which can contain many different types of objects, such as Oracle Dictionary segments, temporary segments, rollback segments, or user segments (that is, tables and indexes).

For more information on the concepts behind partial restore and complete recovery, see [Partial Restore and Complete Recovery \[Page 127\]](#).

SAPDBA supports restoring the database after the loss of the following data files:

- SAP tablespaces (PSAP<name>D/I or PSAP<DBSID>)
- System tablespace (SYSTEM)
- Rollback tablespace (PSAPROLL)
- Temporary tablespace (PSAPTEMP)



The SAPDBA option *Partial Restore and Complete Recovery* only enables recovery of the database up to the **current time**. For more information if you want to restore an older version of the database or perform a point-in-time recovery, see [Performing a Full Restore and Point-In-Time Recovery with SAPDBA \[Page 335\]](#).

Prerequisites

You have replaced the defective disk or otherwise corrected the initial problem.

You have also recreated the damaged file system or raw device at operating system level.

Process Flow

1. If required, you [change the options for restore and recovery \[Page 334\]](#).
2. You [check the database \[Page 324\]](#).
3. You [find the backup files \[Page 326\]](#).
4. You [restore the backup files \[Page 327\]](#).
5. If relevant, you [find the incremental backups \[Page 328\]](#).
6. If relevant, you [restore the incremental backups \[Page 330\]](#).

24.9 Database Restore and Recovery with SAPDBA

7. You [find the offline redo logs \[Page 331\]](#).
8. You [restore the offline redo logs \[Page 332\]](#).
9. You [recover the database \[Page 333\]](#).



If preferred, you can perform [automatic recovery \[Page 334\]](#). This automatically performs all the steps listed above. You do not have so much control with this method.

We normally recommend you to perform the steps individually, so that you can more closely monitor the restore and recovery, and intervene if necessary.

For more information, see the Oracle documentation.



Checking the Database

Use

You check the database when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database. This procedure checks the database for critical errors that might prevent production operation.

SAPDBA analyzes the appropriate V\$ database tables to check:

- 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 Oracle [database monitor \[Ext.\]](#) in the SAP System.

Since some of the V\$ tables are only updated when the database is restarted, SAPDBA differentiates between the following:

- Online quick check

This check is performed with the database open. However, you cannot then recover the database with SAPDBA because the information obtained from this check is not

24.9 Database Restore and Recovery with SAPDBA

necessarily current. As a result, if you have only performed the online quick check, the *Check database* function is set to status *not finished*.

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.

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.

- **Offline safe check**

This is a comprehensive check and is required if you want to later restore and recover the database. Before a safe check, SAPDBA switches the database offline then remounts it. 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.

Prerequisite

- You want to check the database after a problem has occurred.
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- The database can be either online or offline, but a safe check is only possible if the database is offline.

Procedure

1. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Check database* in SAPDBA.
SAPDBA displays the restore and recovery options.
2. If required, [change the restore and recovery options \[Page 334\]](#).
3. Enter y to continue.
SAPDBA checks the control files, data files, links, and bad entries in the control file, then asks whether you want to perform an offline safe check.
4. To perform a safe check, enter y.
If the database is online, SAPDBA warns you that the database will be shut down immediately for the safe check.
5. To shut down the database and continue with the safe check, enter y.
SAPDBA shuts down the database and comprehensively checks the database.

Result

SAPDBA reports any problems that it finds, such as a missing data file.

What happens at the end of the check depends on whether SAPDBA finds any errors:

- If the check was successful (that is, there were no errors), SAPDBA reopens the database. It also sets the *Status* of the check in the menu to *Finished* and the *Status* of all subsequent steps to *not allowed*.
- If errors are found, the database remains offline and SAPDBA does the following:
 - If defective data files are found, SAPDBA creates an internal list of the damaged files. Based on this list, SAPDBA determines how to restore and recover the database.

24.9 Database Restore and Recovery with SAPDBA

- If the control files or redo log files are in a critical condition for the recovery, SAPDBA terminates the database check. You cannot solve this type of error with SAPDBA. You need to take appropriate action using Oracle tools. For more information, see the Oracle documentation and [Recovery with SQLPLUS \[Page 487\]](#). Once you have removed the error, you can restart the database check.

To continue with the partial restore and complete recovery, perform [Finding the Backup Files \[Page 326\]](#).



Finding the Backup Files

Use

You find the backup files when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

SAPDBA attempts to locate the backup files by scanning previous BRBACKUP logs (of a specified period) 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 way.

The information on the location of these files is important for the subsequent phases of the recovery process.

Prerequisite

- You have [checked the database \[Page 324\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Find backup files* in SAPDBA.
2. Choose *Start finding backup files*.

If you used a non-SAP backup program for the backup, SAPDBA also attempts to obtain the corresponding information from this program.

3. Define the period for which you want SAPDBA to search for the backup files. The default value is the previous 30 days.

SAPDBA uses BRBACKUP logs to find the backup files.

If SAPDBA finds backup files, it determines the necessary log sequence number by finding the most recent BRBACKUP file for each lost file and then selecting the smallest of the respective log sequence numbers.

4. Choose one of the following to display the backup files found by SAPDBA:

- *Show the list of damaged files*

SAPDBA lists the lost or damaged 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 or disk.
- backed up by <name of external backup program>

24.9 Database Restore and Recovery with SAPDBA

The file was backed up using the specified program. This comment appears when the parameter [backup_util_name \[Page 198\]](#) 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 or damaged files for which you would like to see the available backup files. Each file that has been lost can have several backup files.

5. Select a backup file for the restore by choosing *Select a backup file for restore*.

This lets you change the SAPDBA proposal for which backup file is restored. The file that is finally selected for the restore process is flagged with `(SELECTED FOR RESTORE)`.

6. Select a BRBACKUP run for the restore by choosing *Select a BRBACKUP run for restore*.

SAPDBA suggests the newest backup found 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 the restore 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 given:

- 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 that 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.

7. Choose *Return* to continue with the recovery.

Result

SAPDBA sets the *Status* of *Find backup files* to *finished*.

To continue with the partial restore and complete recovery, perform [Restoring the Backup Files \[Page 327\]](#).



Restoring the Backup Files

Use

You restore the backup files when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

The restore described in this procedure is part of the partial restore and complete recovery. That is, you must perform a recovery afterwards. To perform a restore independently of a

24.9 Database Restore and Recovery with SAPDBA

recovery, see the other menu options in [Database Restore and Recovery with SAPDBA \[Page 321\]](#).



Restoring a data file overwrites existing data files, since files are restored to their original location.

Prerequisites

- You have [found the backup files \[Page 326\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- During a restore, the SAP tool [BRRESTORE \[Page 395\]](#) physically restores the backup. If space problems occur during a restore of data files, you must use operating system tools to provide the required space.

Procedure

6. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Restore backup files* in SAPDBA.
7. If required, choose *Specify restore parameters* to change the restore parameters:
 - *BRBACKUP Profile*
This is normally [init<DBSID>.sap \[Page 445\]](#).
 - *Backup utility parameter file*
This is normally [init<DBSID>.utl](#). For more information, see [Disaster Recovery \[Page 123\]](#).
 - *Language*
Language of BRRESTORE messages.
3. Choose *Start restore of backup files* to start the restore.
SAPDBA checks whether the files to be restored are still available. SAPDBA warns you that they will be overwritten.
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.
SAPDBA starts the SAP tool BRRESTORE to restore the files.
4. Choose *Return* to continue with the recovery.

Result

When the backup files have been successfully restored, SAPDBA sets the *Status* of *Restore backup files* to *finished*.

To continue with the partial restore and complete recovery:

- If there are incremental backups available, perform [Finding Incremental Backups \[Page 328\]](#). However, if you do not want to restore incremental backups, you can skip the steps involving incremental backups.
- Otherwise, perform [Finding the Offline Redo Log Files \[Page 331\]](#).



Finding Incremental Backups

24.9 Database Restore and Recovery with SAPDBA

Use

You find the incremental backup files when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database, but **only** when you have [incremental backups \[Page 83\]](#) available.



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.

SAPDBA attempts to locate the incremental backups by scanning previous BRBACKUP logs (of a specified period). If the backup was performed with an external backup program, SAPDBA attempts to establish a link to that backup program and find the incremental backups in this way.

The information on the location of these backups is important for the subsequent phases of the recovery process.



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*.

Prerequisite

- You have [restored the backup files \[Page 327\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- You cannot use incremental backups if either of the following applies:
 - Data files were first restored from partial backups or whole backups. Incremental backups are based on full backups (level 0), so must always follow a restore using a full backup.
 - Backups were restored earlier in another *Partial restore and complete recovery* operation, but the recovery phase was never completed. 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 restore fails then you either chose an incorrect incremental backup, or a recovery with incremental backups is not possible.

Procedure

5. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Find incr. backup* in SAPDBA.

6. Choose *Find appropriate incremental backup runs*.

SAPDBA uses BRBACKUP logs to find the backups.

If SAPDBA finds incremental backups, it determines the necessary log sequence number.

24.9 Database Restore and Recovery with SAPDBA

7. Choose one of the following to display details of the incremental backup runs found by SAPDBA, choose *Show list of found incr. backup runs*.
SAPDBA displays the incremental backup runs that it has found.
8. Select a BRBACKUP run for the restore by choosing *Select an incr. backup run for restore*.
9. To check the selected incremental backup, choose *Show selected incr. backup run*.
10. Select *Return* to continue with the recovery process.

Result

SAPDBA sets the *Status* of *Find Incr. Backup* to *finished*.

To continue with the partial restore and complete recovery, perform [Restoring Incremental Backups \[Page 330\]](#).



Restoring Incremental Backups

Use

You restore the incremental backup files when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

The restore described in this procedure is part of the partial restore and complete recovery. That is, you must perform a recovery afterwards. To perform a restore independently of a recovery, see the other menu options in [Database Restore and Recovery with SAPDBA \[Page 321\]](#).



Restoring a data file overwrites existing data files, since files are restored to their original location.

Prerequisites

- You have [found incremental backups \[Page 328\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).
- During a restore, the SAP tool [BRRESTORE \[Page 395\]](#) physically restores the backup. If space problems occur during a restore of data files, you must use operating system tools to provide the required space.

Procedure

1. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Restore incr. backup* in SAPDBA.
2. If required, choose *Specify restore parameters* to change the default restore parameters.
 - *BRBACKUP Profile*
This is normally [init<DBSID>.sap \[Page 445\]](#).
 - *Backup utility parameter file*
This is normally [init<DBSID>.utl](#). For more information, see [Disaster Recovery \[Page 123\]](#).
 - *Language*

24.9 Database Restore and Recovery with SAPDBA

Language of BRRESTORE messages.

3. Choose *Start restore incr. backup run* to restore the incremental backup.

SAPDBA starts BRRESTORE to restore the files.

4. Choose *Return* to continue with the recovery.

Result

When the backup files have been successfully restored, SAPDBA sets the *Status* of *Restore backup files* to *finished*.

To continue with the partial restore and complete recovery, perform [Finding the Offline Redo Log Files \[Page 331\]](#).



Finding the Offline Redo Log Files

Use

You find the offline redo log files when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

SAPDBA attempts to locate valid backups of the offline redo log files by scanning the summary BRARCHIVE log (`arch<DBSID>.log`). If the backup was performed with an external backup program, SAPDBA attempts to establish a link to that backup program and find the offline redo logs in this way.

Prerequisites

- You have [restored the backup files \[Page 327\]](#). If relevant, you have [restored incremental backups \[Page 330\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

1. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Find offline redo logs* in SAPDBA.
2. Choose *Find offline redo logs* so that SAPDBA can make available the offline redo log files belonging to the lost data files.

To find the backed-up offline redo log files, SAPDBA uses the BRARCHIVE summary log. To find the offline redo logs, SAPDBA searches the directory where Oracle archives the online redo log files.

SAPDBA highlights any gaps in the redo log sequence.



If the sequence of the redo log files required for recovery is interrupted, you cannot continue with the recovery. A partial recovery with the SAPDBA menu options in *Partial restore and complete recovery*, but only with loss of data.

3. Choose *Show list of found offline redo logs* to display the list of found offline 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 displays the following log sequence numbers for each instance (thread):

24.9 Database Restore and Recovery with SAPDBA

- 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 normally match.



Only change the log sequence number proposed by SAPDBA in exceptional cases. Only experienced database administrators should attempt this.

4. If necessary, choose *Select instance and change current first required log* to change the log sequence number of the first offline redo log file for the restore.



You must get offline redo log files of other instances that 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.

5. Choose *Return* to continue with the recovery process.

Result

SAPDBA sets the *Status* of *Find offline redo logs* to *finished*.

To continue with the partial restore and complete recovery, perform [Restoring the Offline Redo Log Files \[Page 332\]](#).



Restoring the Offline Redo Log Files

Use

You restore the offline redo log files when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

During a restore, BRRESTORE restores backups of the offline redo log files. If SAPDBA determines that the archiving directory <SAPDATA_HOME>/saparch does not have enough space to restore all the necessary redo log files, the redo logs that have already been used are deleted and the next required redo logs are restored during the subsequent recovery.

Prerequisites

- You have [found the offline redo log files \[Page 331\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

8. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Restore offline redo logs* in SAPDBA.

SAPDBA displays the following information:

- The log sequence number of the first offline redo log file to be restored
- The offline redo log files that were found

24.9 Database Restore and Recovery with SAPDBA

- The maximum size of the offline redo log files
 - The configured restore options, which you can change.
2. If required, choose *Specify restore parameters* to change the restore parameters. However, this is not usually necessary since the parameters are based on the backup logs.
 3. Choose *Start restore of offline redo logs*.



You have to use this procedure when the recovery requires offline redo log files that are no longer in the archiving directory. You cannot start the recovery until the necessary archived redo logs are restored, either now or continuously during the recovery.

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*).

4. Choose *Return* to continue with the recovery.

Result

When the offline redo log files have been successfully restored, SAPDBA sets the *Status* of *Restore offline redo logs* to *finished*.

To continue with the partial restore and complete recovery, perform [Recovering the Database \[Page 333\]](#).



Recovering the Database

Use

You recover the database when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

Prerequisites

- You have [restored the offline redo log files \[Page 332\]](#).
- You have configured and started SAPDBA. Refer to [Getting Started with SAPDBA \[Page 185\]](#).

Procedure

Choose *Restore/recovery* → *Partial restore and complete recovery* → *Recover database* in SAPDBA to recover the database.

SAPDBA tells you:

- When the recovery started
- Whether the database can be recovered in one step
- Which scripts were created to start the recovery

If there was not enough disk space available when you restored the offline redo logs, they 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.

24.9 Database Restore and Recovery with SAPDBA

Result

When the recovery has finished, SAPDBA opens the database again and sets the *Status* of *Recover database* to *finished*.

The partial restore and complete recovery is now complete.



After the recovery, you can use the SAPDBA function *Show/Cleanup* → *Show log files/profiles* to check the SAPDBA log file `<timestamp>.rcv` in directory `<SAPDATA_HOME>/sapreorg`. For more information, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).



Recovering the Database Automatically

Use

You can recover the database automatically when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database. If a data file is lost or damaged, this procedure automatically selects the last valid backup of the relevant data file and the appropriate redo logs.



Automatic recovery does **not** support a recovery with incremental backups.



We normally recommend you to perform the restore and recovery steps individually, so that you can more closely monitor the restore and recovery, and intervene if necessary.

Procedure

1. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Automatic Recovery* in SAPDBA to recover the database automatically.
After each step, SAPDBA asks whether you want to continue the recovery.
2. Respond to the SAPDBA prompts.
3. If you use backups to tape for the recovery, mount the tapes with the correct labels when SAPDBA prompts you.



If the automatic recovery terminates, the SAPDBA menu *Partial Restore and Complete Recovery* appears again. The status display tells you which steps ran successfully (status *finished*). You must perform any unsuccessful steps (that is, with *Status* set to *not finished*) manually after you have solved the problem that led to the termination.

Result

When the recovery has finished, SAPDBA opens the database again and sets the *Status* of *Recover database* to *finished*.



Changing the Restore and Recovery Options

Use

24.9 Database Restore and Recovery with SAPDBA

You can change the options when using SAPDBA to perform a [partial restore and complete recovery \[Page 321\]](#) of your Oracle database.

Procedure

1. Choose *Restore/recovery* → *Partial restore and complete recovery* → *Options* in SAPDBA.
SAPDBA displays the current option values.
2. Set the parameters as required:
 - *Use incremental backups for recovery*
If incremental backups are available and you do not want to use them during the recovery for some reason, set this to *no*.
 - *Recovery by reorganization (for index and empty data files)*
If you want the recovery to be performed by reorganization instead of by applying the redo log files, set this to *yes*.
 - *Use backup from external backup tool to restore files*
If you have an external backup tool, set this to *yes*.
 - *Use inquire function of backint (relying on BR* log information)*
This option only appears if you have set the previous option to *yes*. It means that BACKINT is used to find the files that have been saved as an external backup tool, so that you can decide if you want to use these for the restore.
3. Choose *Return* to continue with the restore or recovery.

24.9.2 Performing a Full Restore and Point-In-Time Recovery with SAPDBA

Use

This section tells you how to use SAPDBA to perform a full restore and point-in-time recovery of your Oracle database. You typically use this after failures:

- Caused by user errors, such as logical errors
- During maintenance, such as upgrade errors or data transfer errors

Prerequisites

The following data must be available:

- The [BRBACKUP logs \[Page 377\]](#) and the [BRARCHIVE logs \[Page 392\]](#)
- The data file backups
- All offline redo log files between the data backup and the reset point in time

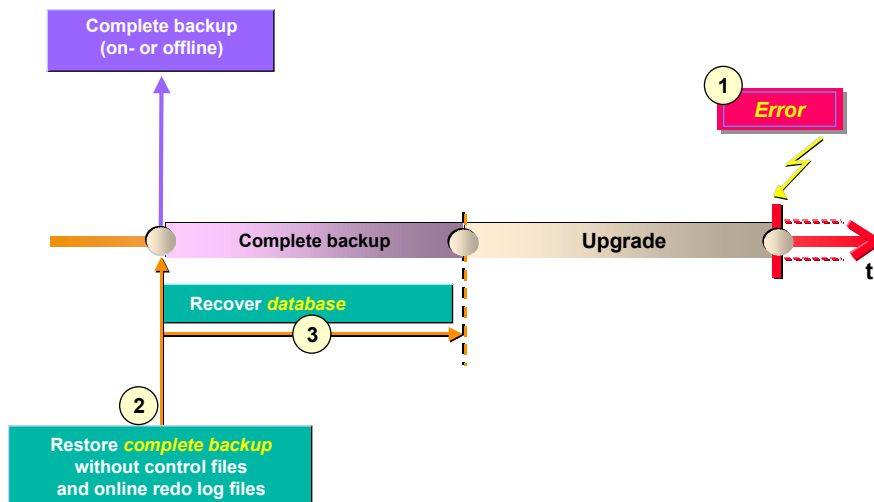
Features

The SAPDBA 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.

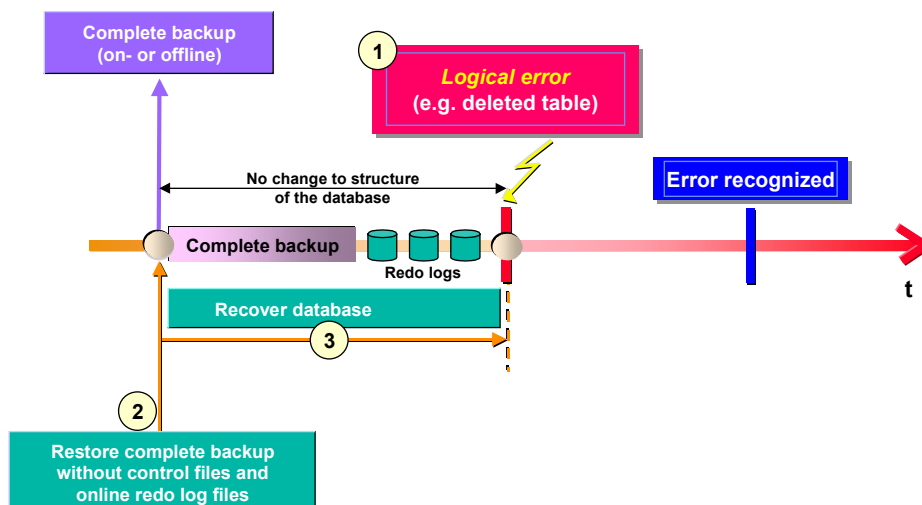
24.9 Database Restore and Recovery with SAPDBA



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.

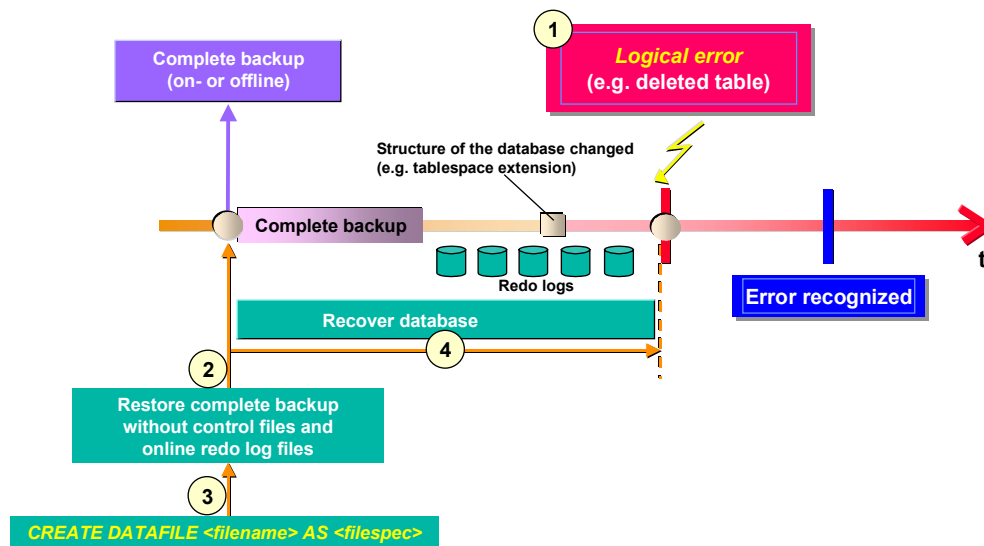


In the first step, SAPDBA restores the last complete backup without control files and online redo log files, and in the second step imports the redo log files and recovers the database.

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.

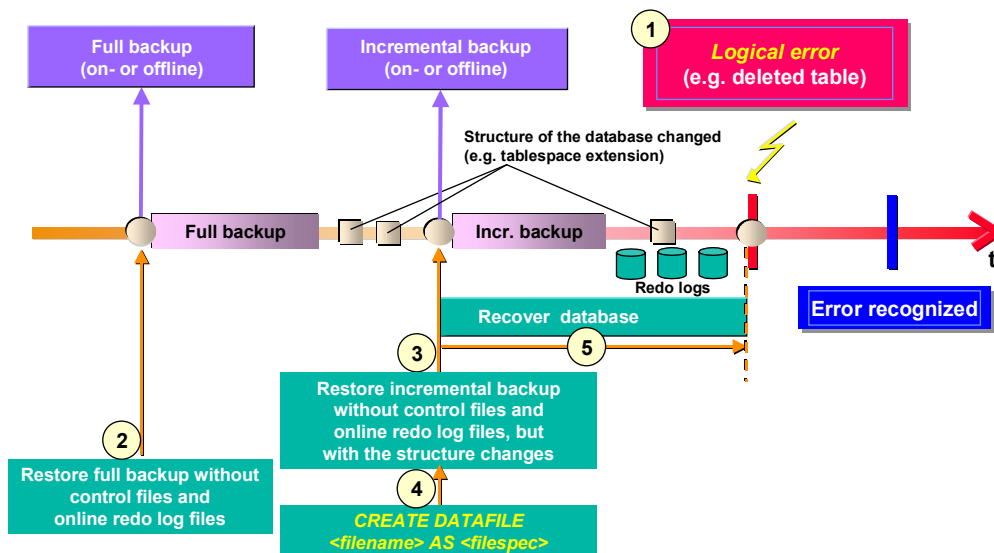
24.9 Database Restore and Recovery with SAPDBA



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.

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.

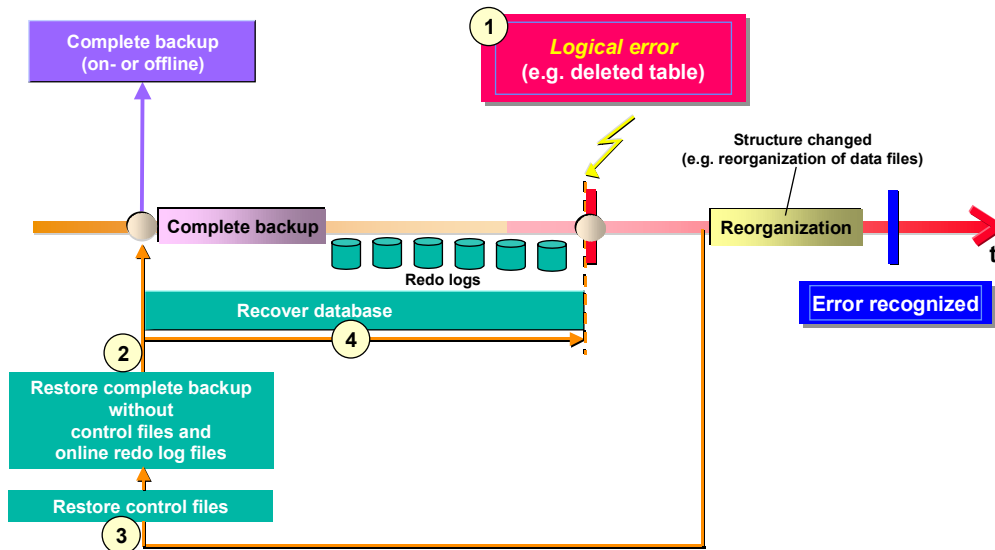


In the first step, SAPDBA restores the last complete backup without control files and online redo log files, and in the second step restores the last incremental backup before the error. The changes to the structure are included in the incremental backup. In the third step, the structural changes made 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

24.9 Database Restore and Recovery with SAPDBA

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.



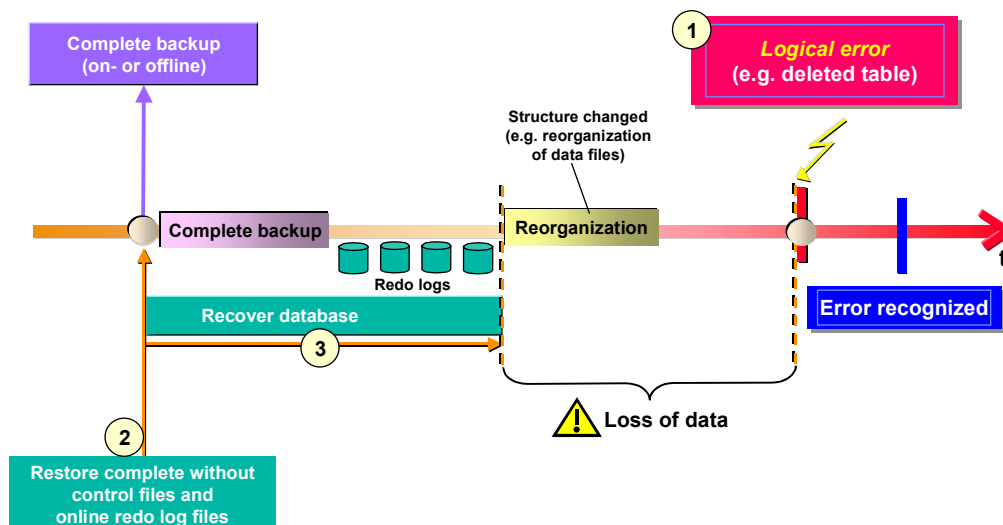
In the first step, SAPDBA restores the last complete backup without control files and online redo log files, in the second step restores the control files as they were before the reorganization. During the reorganization the control files were backed up in the directory <SAPDATA_HOME>/sapreorg/. In the 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.



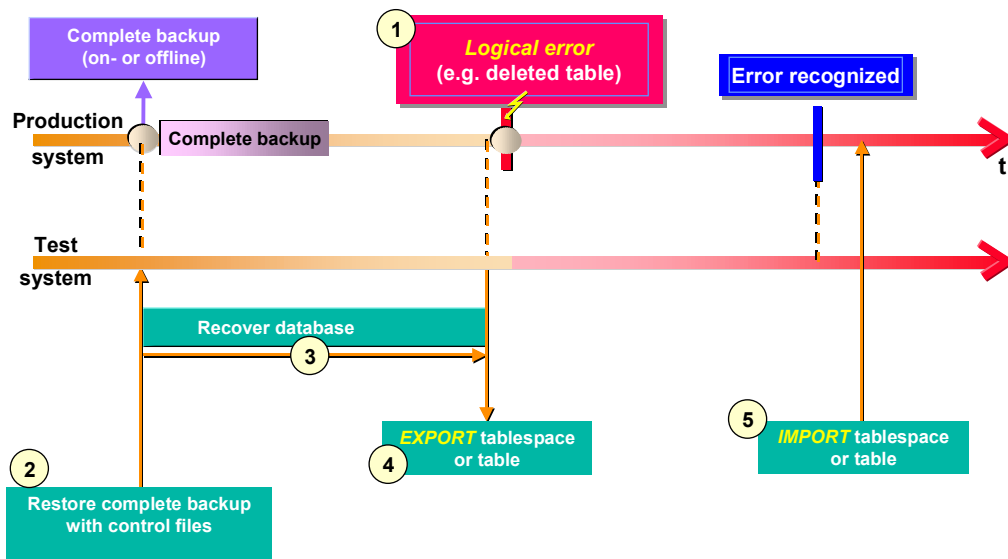
24.9 Database Restore and Recovery with SAPDBA

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	<i>NOMOUNT MOUNT OPEN</i>
	RESTORE/RECOVER	<i>allowed allowed</i>
		<i>Current setting</i>
A	- Select a backup of type	<i>Full backup whole backup</i>
b	- Select incremental backup run	<i>(Only for selected full backup)</i>
c	- Recover until	<i>now point in time</i>
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*.

24.9 Database Restore and Recovery with SAPDBA

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

<SAPDATA_HOME>/sapreorg/<timestamp>.rsn (for *Recover until now*) or
<SAPDATA_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 Log Files and Their Contents \[Page 205\]](#).



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.

24.9.3 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 \[Page 502\]](#) and [Actions after a Partial Recovery \[Page 504\]](#).

- *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.

24.9 Database Restore and Recovery with SAPDBA

The SQLPLUS command for recovering the data is as follows (be sure to read the corresponding Oracle documentation):

```
SQLPLUS> 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
<SAPDATA_HOME>/sapreorg/<timestamp>.rst. To access the file, select the SAPDBA function *Show/Cleanup* → *Show log files/profiles*.

See [SAPDBA Log Files and Their Contents \[Page 205\]](#).

24.9.4 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 and Recovery Options \[Page 334\]](#).

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. For more information, see [Partial Restore and Complete Recovery with SAPDBA \[Page 323\]](#).

Check the log file written by SAPDBA <SAPDATA_HOME>/sapreorg/<timestamp>.dba. To do this, select the SAPDBA function *Show/Cleanup* → *Show log files/profiles*.

See [SAPDBA Log Files and Their Contents \[Page 205\]](#).

24.9.5 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.

24.9 Database Restore and Recovery with SAPDBA



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 \[Page 341\]](#)). 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>
d	- 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<DBSID>.ora` and `init<DBSID>.dba`
7. BRBACKUP/BRARCHIVE profile `init<DBSID>.sap`
8. Main SAPDBA log `reorg<DBSID>.log` and the structure log `struct<DBSID>.log` (if it exists, also the script `create_raw_links`)
9. Summary BRBACKUP/BRARCHIVE log
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>`.



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 [-n|-number \[Page 372\]](#) for BRRESTORE.



Note that logs and profiles can only be restored from certain types of backup (device types `tape`, `pipe`, `tape auto`, `pipe auto`) (see [-d|-device](#)

24.9 Database Restore and Recovery with SAPDBA

[\[Page 367\]](#)). 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).
- 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*)
 - 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.

24.10 SAPDBA Command Options

See [Changing the Restore and Recovery Options \[Page 334\]](#).

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 `<SAPDATA_HOME>/sapreorg/<timestamp>.dba`. Select the SAPDBA function *Show/Cleanup* → *Show log files/profiles*.

See [SAPDBA Log Files and Their Contents \[Page 205\]](#).

24.10 SAPDBA Command Options

Use

You can run SAPDBA using a command option. In this case, 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.



We **strongly recommend** you to use [BRCONNECT commands \[Page 429\]](#) rather than the equivalent SAPDBA ones. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Prerequisites

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

Features

There are the following command options for SAPDBA:

- [sapdba -analyze \[Page 345\]](#)
- [sapdba -check \[Page 347\]](#)
- [sapdba -check_db_open \[Page 348\]](#)
- [sapdba -checkopt \[Page 348\]](#)
- [sapdba -cleanup \[Page 350\]](#)
- [sapdba -delete \[Page 350\]](#)
- [sapdba -export \[Page 350\]](#)
- [sapdba -fill_tables \[Page 351\]](#)
- [sapdba -next \[Page 351\]](#)

- [sapdba -r \[Page 352\]](#)
- [sapdba -shutdown\(_abort\) \[Page 353\]](#)
- [sapdba -startup\(_dba\) \[Page 353\]](#)
- [sapdba -statistics \[Page 353\]](#)
- [sapdba -version \[Page 354\]](#)

Activities

1. You choose a SAPDBA command option
2. You view the [return code for the command option \[Page 355\]](#).

24.10.1 sapdba -alter_user

Use

You can use this SAPDBA command with the Oracle database to change database user passwords.



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f chpass \[Page 432\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Syntax

To change the password for a database user, enter the following command:

```
sapdba -alter_user <name>/<pwd>
```

For more information, see [Changing Database User Passwords \[Page 39\]](#).

24.10.2 sapdba -analyze

Use

You can use the SAPDBA command `sapdba -analyze` with the Oracle database to:

- Analyze the space statistics in a table, tablespace, or list of tablespaces.
- Update or recreate statistics, taking into account the entries in the control table DBSTATC. This is phase II of the update statistics functionality. Phase I is [sapdba -checkopt \[Page 348\]](#).



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f stats \[Page 436\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.



This command and `sapdba -checkopt` are equivalent to the newer integrated approach using [sapdba -statistics \[Page 353\]](#). We recommend you to use the new approach.

24.10 SAPDBA Command Options

For a comparison of the old and new approaches, see [Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#).

For more information on the new approach, see [Updating Statistics using Integrated Approach with SAPDBA \[Page 223\]](#).

The logs for this command are stored in the directory `sapcheck/<timestamp>.aly`.

Syntax

Detailed Command Syntax

```
sapdba
-u system/<password>
-analyze <tablespace> | <tablespace list> | <table> | <keyword> |
[ -method E | C ]
[ -option P<n> | R<m (*1000)> ]
[ -time <days> ]
[ -T <hours> ]
```

SAPDBA Command Options

Command option	Meaning
-u	Database system and password
-analyze	Database objects and key words (see below) for the operation
-method	Analysis method, estimate E or compute C (default is estimate)
-option	Option for the analysis method (only for the estimate method E, for which the default option is P10)
-time	Updates all statistics older than <days> days
-T	Restricts the operation to <hours> hours

Key Words

Key word	Meaning
DBSTATCO	All tables for which the TODO flag is set in the control table DBSTATC and all tables and indexes without statistics (that is, NOOPTSTAT is executed implicitly).
NOOPTSTAT	All tables and indexes without statistics (excluding pooled and cluster tables and tables in the control table DBSTATC with active = N or R).

Analyze Space Statistics in a Table

To analyze space statistics, you enter a command such as the following:

```
sapdba -analyze PSAPBTABD -method C -T 1
```

This command analyzes the statistics for the tablespace PSAPBTABD using the compute method, and sets a time limit of one hour for the operation.



The analysis can be time-consuming and resource-intensive, depending on the hardware used. We recommend you to limit the number of records to be analyzed. For example, analyze selected tables or tablespaces rather than

the entire database. You can also use the estimate method to reduce the time taken.

`sapdba -analyze` produces a report as follows:

- For tablespaces
 - Total space, freespace, allocated space, used space
 - A list of the 20 tables or indexes with the most unnecessarily allocated space, showing unused space, never-been-used space, and used space
- For tables
 - Empty space, never-been-used space, used space
 - Total space, used-by-btree space, used space, deleted space

You can use the information produced to manually optimize the storage of the table, index or tablespace during a reorganization. For more information, see [Checking Statistics for a Reorganization \[Page 245\]](#). SAPDBA can also perform the optimization for you automatically during a reorganization.

Update Statistics

To update statistics, you normally enter the following command:

```
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.

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.

For more information on update statistics, see:

- [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#)
- [Update Statistics: Internal Rules with SAPDBA \[Page 227\]](#)

24.10.3 sapdba -check

Use

You can use this SAPDBA command with the Oracle database to check the database system, as follows:

- 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 our experience of SAP database administration

24.10 SAPDBA Command Options



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f check \[Page 431\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Prerequisites

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. In the following example, the file `<file_name>` contains the password:



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

Syntax

Use the command `sapdba -check -e <no._of_extents>` to adjust the default number of extents (2), for checking the tablespace overflow. However, we recommend you to make individual adjustments to the database system check in the table `DBCHECKORA` (transaction DB17) using the Computing Center Management System (CCMS) of the SAP System.

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

Result

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

See also:

- [Checking the Database System \[Ext.\]](#)
- [Configuring the Database System Check \[Ext.\]](#)

24.10.4  **sapdba -check_db_open**

You can use this SAPDBA command with the Oracle database to check 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).



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f dbstate \[Page 430\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

24.10.5  **sapdba -checkopt**

Use

You can use this SAPDBA command with the Oracle database to determine whether new statistics are needed. It enters 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.



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f stats \[Page 436\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

This is phase I of the update statistics functionality. Phase II is [sapdba -analyze \[Page 345\]](#).



This command and `sapdba -analyze` are equivalent to the newer integrated approach using [sapdba -statistics \[Page 353\]](#). We recommend you to use the new approach.

For a comparison of the old and new approaches, see [Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#).

For more information on the new approach, see [Updating Statistics using Integrated Approach with SAPDBA \[Page 223\]](#).

Syntax

You normally enter the following command:

```
sapdba -checkopt PSAP%
```

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

Here is the detailed command syntax:

```
sapdba
-u system/<password>
-checkopt <tablespace> | <tablespace list> | <table> | <keyword>
[ -T <hours> ]
force
```

SAPDBA Command Options

Command option	Meaning
-u	Database system and password
-checkopt	Database objects and key words for the operation
-T	Restricts the operation to <hours> hours
-force	Resets the <code>TODO</code> flags in the control table <code>DBSTATC</code> regardless of the current situation



If you enter `DBSTATC_TAB` for <keyword>, all tables entered in the control table `DBSTATC` are used.

SAPDBA chooses the correct analysis method and option and enters these 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 logs for this command are stored in the directory `sapcheck/<timestamp>.opt`.

See also:

24.10 SAPDBA Command Options

- [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#)
- [Update Statistics: Internal Rules with SAPDBA \[Page 227\]](#)

24.10.6 sapdba -cleanup

You can use this SAPDBA command with the Oracle database to remove logs and directories according to the default values set in the SAPDBA profile.



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f cleanup \[Page 432\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

For more information, see [Initialization Profile init<DBSID>.dba \[Page 212\]](#) and [Deleting Log and Trace Files \[Page 198\]](#).

24.10.7 sapdba -delete

Use

You can use this SAPDBA command with the Oracle database to delete statistics used by the cost-based optimizer (CBO).



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f stats \[Page 436\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Syntax

To delete statistics, for one or more tablespaces or for a particular table enter the following command:

```
sapdba -delete <tablespace(s)> | <table>
```

To 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 the `DBSTATC` control table):

```
sapdba -delete HARMFUL
```

See also:

[sapdba -checkopt \[Page 348\]](#)

[Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#)

[Updating Statistics using Optimizer Control with SAPDBA \[Page 226\]](#)

[Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#)

24.10.8 sapdba -export

Use

You can use this SAPDBA command with the Oracle database to export objects. This command is important for testing tables and indexes for corrupted Oracle blocks. No export dump file is generated.

Prerequisites

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, that is, `/dev/null`.

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.

Syntax

```
sapdba -export <tablespace(s)/table>
```

See also:

[Export and Import of Database Objects \[Page 285\]](#)

24.10.9 sapdba -fill_tables

You can use this SAPDBA command with the Oracle database to fill the tables *DBSTATC*, *DBCHECKORA* and *DBAFID* with standard data. This does not overwrite any existing entries.

24.10.10 sapdba -init_sap_connect

Use

You can use this SAPDBA command with the Oracle database to initialize various values, especially when you install the database.

Syntax

To initialize the SAP connect, enter the following command:

```
sapdba -init_sap_connect -sapsid <SID1>,<SID2>,...
```

For more information, see [Initializing SAP Connect \[Page 40\]](#).

24.10.11 sapdba -next <tablespace(s)>

Use

You can use this SAPDBA command to adjust the value of the next extent allocated to growing tables in the Oracle database.



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function `-f next [Page 435]` instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

If you are reorganizing a tablespace that grows quickly (for example, a tablespace with transaction data), we recommend using this command frequently. SAPDBA adjusts the value of `NEXT` according to the rapid growth of the tablespace. Adjusting `NEXT` prevents the tablespace from reaching the `MAXEXTENTS` limit too quickly.

24.10 SAPDBA Command Options

SAPDBA chooses the new value for NEXT as the **maximum** of the following two values:

- An SAP object is assigned a size category from table DD09L. The NEXT value is determined from table TGORA/IGORA based on the category. If the table has no category assigned in DD09L, this value is not generated.
- SAPDBA chooses the maximum of the following values:
 - The current NEXT value
 - 10% of the total allocated space

SAPDBA compares the new chosen value for NEXT 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 there are no entries for the table in TGORA/IGORA, SAPDBA uses an internal default value. See also [Default Value for NEXT \[Page 257\]](#).

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. For more information on the log, see [SAPDBA Log Files and Their Contents \[Page 205\]](#).

Syntax

The following are examples of the syntax for this command:



- For all SAP tablespaces:
`sapdba -next PSAP%`
- For all SAP index tablespaces:
`sapdba -next PSAP%I`
- For the tablespaces PSAPBTABD and PSAPBTABI:
`sapdba -next PSAPBTAB%`
- For single tablespaces that grow quickly, for example PSAPBTABD (run this command for these tablespaces more often):
`sapdba -next <tablespace_name>`
- For a list of tablespaces (note that there is no space between the tablespace names):
`sapdba -next <TSP_name1>,<TSP_name2>,...`



You **cannot** change the NEXT parameter for the SYSTEM tablespace.

24.10.12 sapdba -r

Use

You can use this SAPDBA command to start reorganization for the Oracle database.

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 \[Page 283\]](#).

Syntax

```
sapdba -r <complete path>/<timestamp>
```

24.10.13 sapdba -shutdown(_abort)

Use

You can use this SAPDBA command to shut down the Oracle database. For more information, see [Shutting Down the Oracle Database System \[Page 219\]](#).



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f dbshut \[Page 434\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Syntax

- The following command shuts down the database cleanly:

```
sapdba -shutdown
```

- The following command shuts down the database immediately **without** ending any running processes cleanly:

```
sapdba -shutdown_abort
```

24.10.14 sapdba -startup(_dba)

Use

You can use this SAPDBA command to start up the Oracle database. For more information, see [Starting Up the Oracle Database System \[Page 218\]](#).



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f dbstart \[Page 430\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Syntax

- The following command starts up the database normally.

```
sapdba -startup
```

- The following command starts up the database in DBA or restricted session mode.

```
sapdba -startup_dba
```

24.10.15 sapdba -statistics

Use

You can use this SAPDBA command to check and update statistics for the Oracle database in a single integrated phase.

24.10 SAPDBA Command Options



We **strongly recommend** you to use the [BRCONNECT \[Page 407\]](#) equivalent function [-f stats \[Page 436\]](#) instead of this SAPDBA command. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.



This command is equivalent to the older two-phase approach using [sapdba –checkpoint \[Page 348\]](#) and [sapdba –analyze \[Page 345\]](#). We recommend you to use the new approach.

For a comparison of the old and new approaches, see [Update Statistics for the Cost-Based Optimizer with SAPDBA \[Page 222\]](#).

For more information on the new approach, see [Updating Statistics using Integrated Approach with SAPDBA \[Page 223\]](#). This also has more information on the commands listed below.

Syntax

Here is the detailed command syntax:

```
sapdba
-statistics ALL | DBSTATCO | <tables>
[ -method E | C | EH | CH
-option Pn | Rn ]
```

The following examples illustrate the use of this command:



- To update **all** database statistics:

```
sapdba -statistics
```

 You can put `-ALL` at the end of this statement, but this does not change the command and is not necessary.
- To **only** update statistics for tables listed in the DBSTATC table:

```
sapdba -statistics DBSTATCO
```
- To update all tables beginning with `psap`:

```
sapdba -statistics psap*
```
- To force SAPDBA to use a particular method or option for update statistics:

```
sapdba -statistics ALL -method E -option P10
```

24.10.16 sapdba -version

Use

You can use this SAPDBA command to display administration information about the current SAPDBA version being used with the Oracle database. The information displayed includes the following:

- Release
- Date of compilation of the current version
- SAPDBA sources
- Development system
- Date of the last release
- Oracle version used for compilation

Prerequisites

Logs written by the SAPDBA commands are stored as follows:

Command	Directory for storage of logs
sapdba -check	<SAPDATA_HOME>/sapcheck/<timestamp>.chk
sapdba -checkopt	<SAPDATA_HOME>/sapcheck/<timestamp>.opt
sapdba -analyze	<SAPDATA_HOME>/sapcheck/<timestamp>.aly
sapdba -next	<SAPDATA_HOME>/sapcheck/<timestamp>.nxt
sapdba -startup(_dba) sapdba -shutdown(_abort)	<SAPDATA_HOME>/sapreorg/<timestamp>.dba

Syntax

sapdba -version

See also:

- [Management of SAPDBA Logs and Profiles \[Page 197\]](#)
- [SAPDBA Log Files and Their Contents \[Page 205\]](#)
- [Return Codes for SAPDBA Command Options \[Page 355\]](#)

24.10.17 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

25 BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT

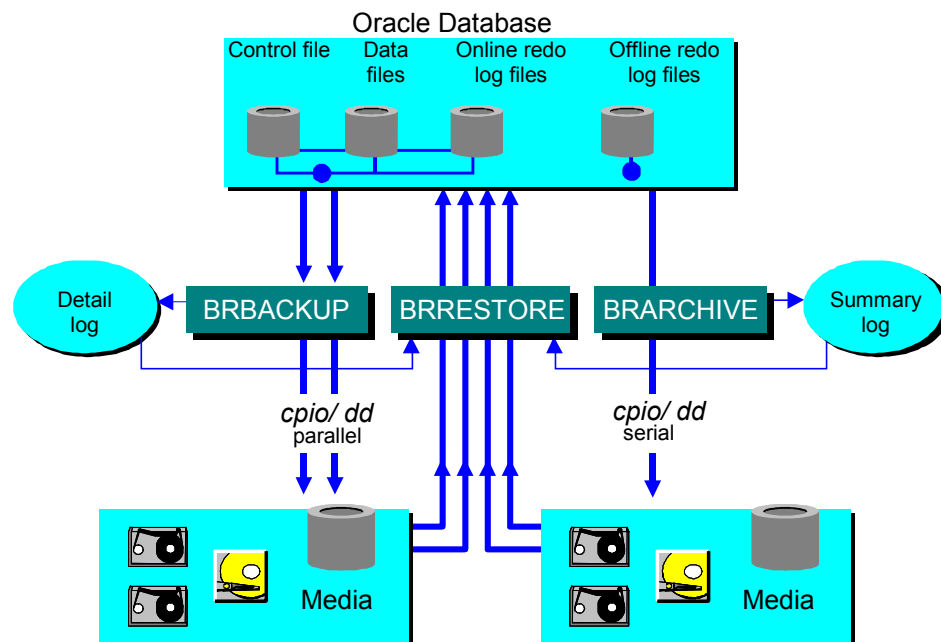
Use

SAP provides the tools BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT to protect the data in your Oracle database. Operating system programs – such as “dd”, “cpio”, or “tar” used directly on UNIX – are too limited and so **not** suitable for this task.

Integration

- The tools are integrated with SAPDBA, which:
 - Calls BRBACKUP, BRARCHIVE, and BRRESTORE
 - Performs automatic database restore and recovery
 - Administers the redo logs
- The tools are available for the operating systems UNIX and Windows NT.
- The tools have their own utility tool, BRCONNECT and [BRTOOLS \[Page 444\]](#).

The following graphic shows the integration of the backup and restore tools:



Features

- [Common Features of BRBACKUP and BRARCHIVE \[Page 357\]](#)
Features common to **both** BRBACKUP and BRARCHIVE
- [BRBACKUP \[Page 360\]](#)
Backup of data files, control files, and online redo log files of the database
- [BRARCHIVE \[Page 383\]](#)
Archiving of offline redo log files

- [BRRESTORE \[Page 395\]](#)

Restore of data files, control files, online redo log files, and offline redo log files

- [BRCONNECT \[Page 407\]](#)

Database administration tasks such as statistics update, check database system, and adapt next extents. BRCONNECT **also** functions as a utility tool in a similar way to BRTOOLS.

25.1 Common Features of BRBACKUP and BRARCHIVE

Use

This section describes the **common** features of [BRBACKUP \[Page 360\]](#) and [BRARCHIVE \[Page 383\]](#). You can call these tools as follows:

- Directly from the UNIX command level, using the SAPDBA menu – see [Database Backup with SAPDBA \[Page 303\]](#), [Backup of the Offline Redo Logs Files with SAPDBA \[Page 313\]](#)
- From the SAP System, using the [Computing Center Management System \(CCMS\) \[Ext.\]](#)

BRBACKUP and 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 or dd for backup to tape, cp or dd for backup to disk if you are working with file systems.
- dd for backup of raw devices on tape or disk

You must configure BRBACKUP and BRARCHIVE, using the parameters in the [initialization profile init<DBSID>.sap \[Page 445\]](#).

The default configurations of both programs often require online user interaction. You can also run the programs without interaction, as described in [Unattended Backup \[Page 114\]](#).

- BRBACKUP saves database objects as follows:
 - 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 \[Page 361\]](#).

- 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.
- BRBACKUP and BRARCHIVE also back up the following files to the volume (when database objects or offline redo log files are backed up) or to the directory named in backup_root_dir for a backup of database files to disk:
 - A copy of profile init<DBSID>.ora, init<DBSID>.dba
 - A copy of profile init<DBSID>.sap
 - The main log of SAPDBA reorg<DBSID>.log, the structure log struct<DBSID>.log

25.1 Common Features of BRBACKUP and BRARCHIVE

- The detail BRBACKUP and BRARCHIVE log
- The summary BRBACKUP and BRARCHIVE log

The logs are saved on every volume (for example, tape) used for the backup. As a result, you can still determine the contents of the volume, even when the database and file system logs from BRBACKUP or BRARCHIVE have been lost.

- [BRCONNECT \[Page 407\]](#) monitors the database during a BRBACKUP process. [BRTOOLS \[Page 444\]](#) is called internally by BRBACKUP and BRARCHIVE. You cannot call BRTOOLS directly yourself.

For more information, see:

- [Supported Backup Media \[Page 358\]](#)
- [Backup with Automatic Tape Changers \[Page 176\]](#)

25.1.1 Supported Backup Media

You can use [BRBACKUP \[Page 360\]](#) or [BRARCHIVE \[Page 383\]](#) for direct backup to the following media:

- Local tape devices:
`backup_dev_type = tape`
- Remote tape devices:
`backup_dev_type = pipe`
- Local disks:
`backup_dev_type = disk`
- Remote disks:
`backup_dev_type = stage`
- 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`

For more information, see [Backup with Automatic Tape Changers \[Page 176\]](#).

- Jukebox and autoloader. These backup devices can be addressed:
 - Locally with `backup_dev_type = tape_box`
 - Remotely with `backup_dev_type = pipe_box`

For more information, see [Backup with Automatic Tape Changers \[Page 176\]](#).

BRBACKUP/BRARCHIVE support:

- Automatic volume management. See [Volume Management \[Page 91\]](#).
- An open interface to interface program BACKINT, to enable backups using external backup programs. See [External Backup Programs \[Page 180\]](#).

Using the parameter [tape_copy_cmd \[Page 477\]](#) you can choose whether files (not on raw devices) are copied from disk to tape with `cpio` or `dd`.

25.1.2 Effects of the Command Options

For more information about command options, see:

- [Command Options for BRBACKUP \[Page 363\]](#)
- [Command Options for BRARCHIVE \[Page 384\]](#)
- [Command Options for BRRESTORE \[Page 398\]](#)
- [Command Options for BRCONNECT \[Page 429\]](#)

Several parameters of the [init<DBSID>.sap profile \[Page 445\]](#) can be overridden by calling BRBACKUP or BRARCHIVE with the appropriate command option.

The following table displays the corresponding profile parameters and the options that override them:

Parameter	Command option	SAP tool
archive_function	-s -save -cs -copy_save (and others)	BRARCHIVE
backup_mode	-m -mode	BRBACKUP
restore_mode	-m -mode	BRRESTORE
backup_type	-t -type	BRBACKUP
backup_dev_type	-d -device	BRBACKUP, BRARCHIVE, BRRESTORE
compress	-k compress	BRBACKUP, BRARCHIVE, BRRESTORE
exec_parallel	-e -execute	BRBACKUP, BRRESTORE
volume_archive	-v -volume	BRARCHIVE
volume_backup	-v -volume	BRBACKUP
util_par_file	-r -parfile	BRBACKUP, BRARCHIVE, BRRESTORE
saveset_members	-s -saveset	BRBACKUP

25.1.3 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.

25.2 BRBACKUP

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.

25.1.4 cpio Error

cpio might report an I/O error:

I/O Error on output.

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.

25.1.5 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.

25.2 BRBACKUP

This SAP tool for the Oracle database enables you to back up database files. For further information about features common to both tools, see [Common Features of BRBACKUP and BRARCHIVE \[Page 357\]](#).

The smallest unit that can be saved with BRBACKUP is a file. You can use BRBACKUP for backing up both files in the database and non-database files and directories. Use the [backup_mode \[Page 449\]](#) from the [Initialization Profile init<DBSID>.sap \[Page 445\]](#) or the command option `brbackup -m|-mode [Page 370]` for this purpose.

For more information, see:

- [Backing up Non-Database Files and Directories \[Page 361\]](#)
- [Backing up Database Files \[Page 361\]](#)
- [BRTOOLS \[Page 444\]](#)

- [Hardware Compression for BRBACKUP \[Page 363\]](#)
- [Logging \[Ext.\]](#)
- [Completion of BRBACKUP Backups \[Page 362\]](#)

25.2.1 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 \[Page 81\]](#).

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` or `full`, all the database data files (and therefore all the tablespaces) and control file are saved. For an offline backup, a file is also saved for every online redo log group.
- 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. However, we do not recommend this procedure. 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`. However, this is really only required for offline backup. It is not normally required because complete offline backups save these files automatically.
- 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 `SAPDATA_HOME` directory and an additional generic specification (for example, `sapdata<n>` directory).

See also [backup_mode \[Page 449\]](#) or [-m|-mode \[Page 370\]](#).

25.2.2 Backing Up Non-Database Files and Directories

Non-database files should only be backed up with BRBACKUP after an SAP upgrade or an Oracle upgrade. 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.

25.2 BRBACKUP

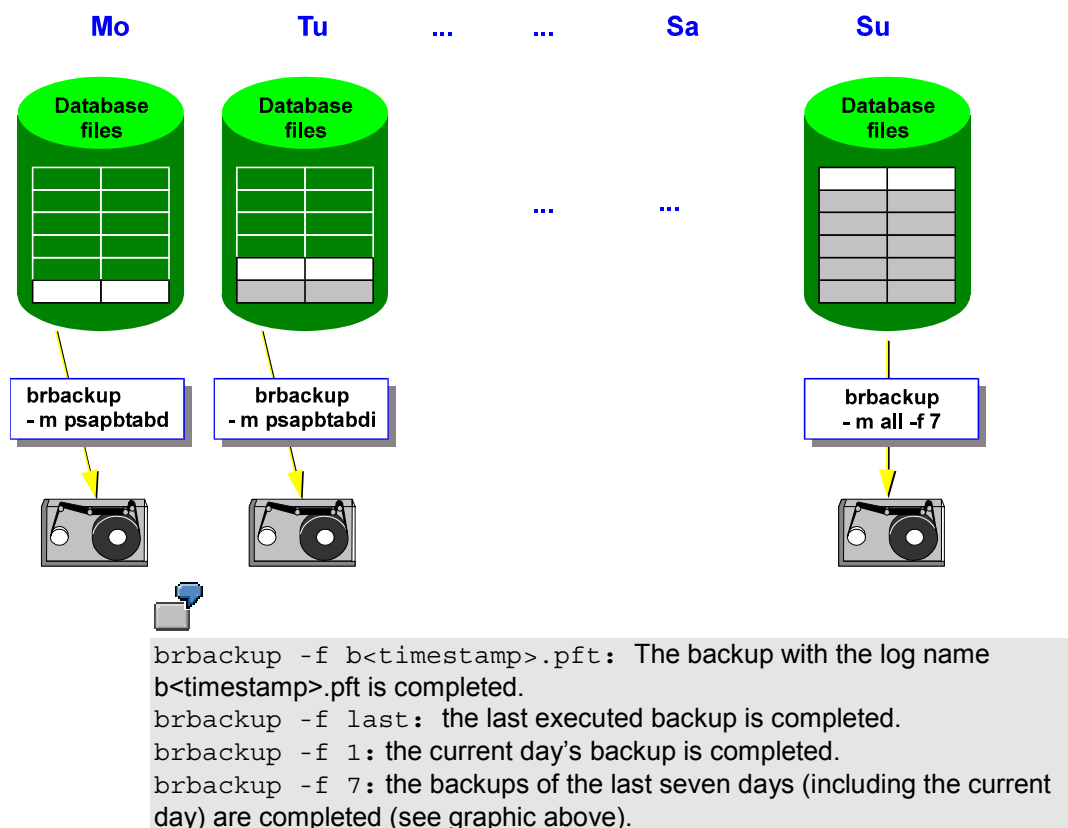
- If you want to back up a large number of non-database files and directories (all the SAP executables and profiles, for example), we recommend carrying out this backup separately.

See also [backup_mode \[Page 449\]](#) or [-m|-mode \[Page 370\]](#).

25.2.3 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 \[Page 369\]](#) you can simply continue the backup. BRBACKUP automatically determines the files that are still to be backed up: target set (defined with the parameter backup_mode or the option -m) minus set backed up (BRBACKUP detailed log). The completion of a backup can also refer to several terminated backups. In this case a new set of files to be backed up can be specified, differing from the original target volume set.
- 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 (for example, 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.



See also:

[-f|-fillup \[Page 369\]](#)

[Completion of BRRESTORE Runs \[Page 396\]](#)

[Partial Backups \[Page 159\]](#)

25.2.4 Hardware Compression for BRBACKUP

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 to be saved 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 \[Page 455\]](#).

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.

For more information, see [Logs for BRBACKUP, BRARCHIVE, BRRESTORE and BRCONNECT \[Page 482\]](#).



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 \[Page 101\]](#).

25.2.5 Command Options for BRBACKUP

This section describes the command options for [BRBACKUP \[Page 360\]](#). If you start BRBACKUP without command options, the values in the [Initialization Profile init<DBSID>.sap \[Page 445\]](#) are used. Unless otherwise specified in the initialization profile `init<DBSID>.sap`, BRBACKUP starts an offline complete backup to a local tape device, with a storage capacity of 1200 MB and no file compression.

For more information about features common to both tools, see [Common Features of BRBACKUP and BRARCHIVE \[Page 357\]](#).

If you use BRBACKUP with command options, these override the corresponding values in the initialization profile. To use the options, you can specify either the letter indicated or the complete word.

```
brbackup
[-a|-archive [<bra_options>]]
[-b|-backup <log_name>|last]
[-bd|-backup_delete <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|stage|stage_copy|stage_
standby|rman_prep|rman_util]
[-db|-delete_backup <log_name>|last]
[-e|-execute <number>]
[-f|-fillup <log name1>[<log name2>,..]|<days>|last]
[-h|-help [version]]
[-i|-initialize [force|show]]
```

25.2 BRBACKUP

```

[-k|-compress no|yes|hardware|only]
[-l|-language E|D]
[-m|-mode
all|<tablespace_name>|all_data|full|incr|<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>]
[-s|-save set 1|2|3|4|tsp|all]
[-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 [use_dbv|only_dbv]]
[-V|-VERSION]

```

[-a|-archive \[Page 364\]](#)

[-b|-backup \[Page 365\]](#)

[-c|-confirm \[Page 366\]](#)

[-d|-device \[Page 367\]](#)

[-e|-execute \[Page 368\]](#)

[-f|-fillup \[Page 369\]](#)

[-h|-help \[Page 369\]](#)

[-i|-initialize \[Page 369\]](#)

[-k|-compress \[Page 370\]](#)

[-l|-language \[Page 370\]](#)

[m|-mode \[Page 370\]](#)

[-n|-number \[Page 372\]](#)

[-o|-output \[Page 372\]](#)

[-p|-profile \[Page 372\]](#)

[-q|-query \[Page 373\]](#)

[-r|-parfile \[Page 373\]](#)

[-t|-type \[Page 374\]](#)

[-u|-user \[Page 375\]](#)

[-v|-volume \[Page 375\]](#)

[-w|-verify \[Page 376\]](#)

[-V|-VERSION \[Page 377\]](#)

**-a|-archive**

This BRBACKUP command option starts BRARCHIVE at the end of a BRBACKUP run.

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 \[Page 384\]](#) and [Initialization Profile init<DBSID>.sap \[Page 445\]](#)



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 -ssd -c
```

See also:

[BRBACKUP and BRARCHIVE Backups in One Run \[Page 116\]](#)



-b|-backup

This BRBACKUP command option backs up a disk 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 or remote disk.
- **last:** BRBACKUP selects the last successful disk backup and copies this to the current backup tape or remote disk.



Backup of the last disk backup with the same parameters as defined in `init<DBSID>.sap`:

```
brbackup -b
```

See also:

[Two-Phase Backup \[Page 109\]](#)



-bd|-backup_delete

This BRBACKUP command option backs up then deletes a disk backup.

Input syntax: `-bd|-backup_delete [<log_name>|last]`

Default: `last`

25.2 BRBACKUP

- **<log_name>**: The BRBACKUP backup to disk with the log name entered in **<log_name>** is copied to the current backup tape or remote disk.
- **last**: BRBACKUP selects the last successful backup to disk and copies this to the current backup tape or remote disk.

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 \[Page 109\]](#)

**-c|-confirm**

This BRBACKUP command option backs up in unattended mode.

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 \[Page 114\]](#).
- 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 \[Page 359\]](#)). 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 \[Page 114\]](#).

If you want to run the SAP utility with, for example, the UNIX utility CRON, you should use the option `-c force`.



-db|-delete_backup

This BRBACKUP command option deletes a disk backup.

Input syntax: `-db [<log_name>|last]`

Default: last

- `<log_name>`: The BRBACKUP disk backup called `log_name` is deleted from disk.
- `last`: BRBACKUP selects the last successful disk backup and deletes it from disk.



-d|-device

This BRBACKUP command option defines the backup device type.

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`: Local disk.
- `tape`: 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 SAP_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

25.2 BRBACKUP

in the parameters `tape_address` or `tape_address_arch`, the drivers for rewinding 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 use this option, 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`: Backup to remote disk. See [backup_dev_type \[Page 447\]](#)
- `rman_util` | `rman_prep`: Backup with the Oracle Recover Manager (RMAN) and an external backup tool. See [backup_dev_type \[Page 447\]](#)

See also:

Parameters in `init<DBSID>.sap`: [backup_dev_type \[Page 447\]](#).

**-e|-execute**

This BRBACKUP command option executes the backup in parallel.

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` | `stage_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 \[Page 459\]](#).



-f|-fillup

This BRBACKUP command option makes backup runs complete.

Input syntax: `-f <log_name1>[,<log_name2,...>] | <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 `<days>` or to the last backup to be run.



-h|-help

This BRBACKUP command option provides help information.

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.



-i|-initialize

This BRBACKUP command option initializes tape volumes.

Input syntax: `-i [force|show]`

Default value: Label checking before initialization.

Use this option to initialize SAP volumes (tapes) or non-SAP volumes. Only tapes of this type can be used for backup with BRBACKUP. See [Volume Management \[Page 91\]](#).

- **-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.
- **-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`. However, we recommend you to only use this option when necessary, since the expiration period is not checked and the `tape_use_count` stored in the volume label is reset to one.

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
Reliant	cpio: Bad header - checksum error. cpio: Not a cpio file.
AIX	Cannot read from the specified input.

25.2 BRBACKUP

	Out of phase.
HP-UX	Out of phase -- get help. End of volume.
TRU64	Not a cpio archive.
Solaris	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**

This BRBACKUP command option sets the compression mode.

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 \[Page 101\]](#)
- **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. For more information, see [Hardware Compression for BRBACKUP \[Page 363\]](#).

Also see [compress_cmd \[Page 455\]](#).

Parameters in `init<DBSID>.sap`: [compress \[Page 454\]](#).

**-l|-language**

This BRBACKUP command option sets the language for messages.

Input syntax: **-l E|D**

Default value: **E**

You can determine whether you want the 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

This BRBACKUP command option defines the file to be backed up.

Input syntax:

```
-m all|all_data|full|incr|<tablespace>|<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 \[Page 111\]](#) (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 -d disk_copy -m all,d:=k:,e:=l:,f:=m:
```

If you do not specify a target drive, all files are copied to the directory defined in the parameter [new_db_home \[Page 461\]](#).

- **all_data**: Back up the files of all tablespaces, except for pure index tablespaces.
- **full**: Full database backup at level 0. See [Complete Backups \[Page 83\]](#).
- **incr**: Incremental backup with RMAN. See full database backup (level 0) in [Incremental Backup \[Page 83\]](#).
- **<tablespace>**: Back up the files of one tablespace.
- **<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 SAPDATA_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.

- **<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.

25.2 BRBACKUP

- **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 \[Page 449\]](#).

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**

This BRBACKUP command option defines the number of tape volumes to be initialized.

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**

This BRBACKUP command option prints extra information to the log file.

Input syntax: `-o dist|time[,time|dist]`

Default value: The BRBACKUP detail log is written in normal form. See [BRBACKUP Detail Log \[Page 378\]](#).

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 \[Page 484\]](#).

**-p|-profile**

This BRBACKUP command option defines the profile name.

Input syntax: `-p <profile>`

Default value: `init<DBSID>.sap`

This profile is contained in directory `<ORACLE_HOME>/dbs` (UNIX) or `<ORACLE_HOME>\database` (Windows NT).

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, specify the complete path.

-q|-query

This BRBACKUP command option sets query mode.

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

This BRBACKUP command option defines the BACKINT parameter file.

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 when 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 name 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, contact the supplier of the external backup program.

Parameters in `init<DBSID>.sap`:

- [mount_par_file \[Page 460\]](#) (for BRBACKUP backups)
- [util_par_file \[Page 480\]](#) (for external backup programs)

25.2 BRBACKUP



-s|-saveset

This BRBACKUP command option defines the number of files in a save set.

Input syntax: `-s 1 | 2 | 3 | 4 | tsp | all`

Default: 1

1, 2, 3, 4: Number of files in a save set

`tsp`: each save set contains all files of a tablespace

`all`: only one save set with all database files is created

The SAP backup library helps to optimize the utilization of quick tape drives by combining multiple data files in save sets. Multiple file access (file multiplexing) maximizes the flow of data (streaming mode).

A save set can contain individual data files, all files of a tablespace, or the complete data backup. The size of the save sets for the backup must be selected according to the tape device. A fast data flow with a minimum save set size is the optimum.



We do not recommend large save sets, since in a restore the complete save set 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 save set” is created including all changed blocks.

For more information, refer to [RMAN Save-Set Grouping \[Page 151\]](#).



-t|-type

This BRBACKUP command option defines the online or offline backup 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 performed for the open database.
- **online_cons**: The backup is carried out whilst the database is open. 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 standby database configuration. For more information, see [Standby Database \[Page 162\]](#).

- **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: BRBACKUP Backup of Database Files \[Page 167\]](#)
- **online_split**: The splitting and saving of the mirror disks is performed while the database is open. This option is only relevant for the split mirror configuration. See [Split Mirror Online Backup \[Page 172\]](#).
- **offline_split**: The database is stopped for the splitting of the mirror disks. The backup of the mirror disks is then done directly afterwards, whilst the database is open. This option is only relevant for the split mirror configuration. See [Split Mirror Offline Backup \[Page 173\]](#).

For more information, see parameter [backup_type \[Page 450\]](#) in `init<DBSID>.sap`.



-u|-user

This BRBACKUP command option defines the user name and password.

Input syntax: `-u [<user>[/<password>]]`

Default value: **system/manager**

User name and password used by the SAP utility program to log on to the database system.

If you only enter `-u`, an interactive query of the user name 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:

```
brbackup -u /
```

In this case, BRBACKUP tries to log on to the database as OPS\$ user (see Oracle documentation and information in the SAP Service Marketplace at <http://service.sap.com/notes>). The OPS\$ user must be defined in the database and have at least SYSOPER authorization and SAPDBA role. The user must also have SYSDBA authorization, if RMAN is to be used. With this method, it is not necessary to specify the password when calling BRBACKUP.



-v|-volume

This BRBACKUP command option defines the tape volumes to be used.

Input syntax: `-v <volume>|<volume list>|SCRATCH`

25.2 BRBACKUP

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 \[Page 91\]](#).

Parameters in `init<DBSID>.sap`: [volume_archive \[Page 480\]](#) or [volume_backup \[Page 481\]](#).



-w|-verify

This BRBACKUP command option verifies the backup after the files have been backed up.

Input syntax: `-w [use_dbv|only_dbv]`

Default: No verification.

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, decompressed (when `compress = yes` was used), read by the check program, and compared with the originals. See also [compress_dir \[Page 455\]](#).

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 \[Page 484\]](#).

use_dbv: Database backup followed by a restoration to a temporary directory and a check of the Oracle block structure with the DBVERIFY tool.

only_dbv: Internal block structure check on the database files with DBVERIFY without backup:

Note that verification approximately doubles the backup time required.

For security reasons, we recommend using this option at least once within the volume expiration period for your complete backups, or still better, once a week. It enables you to detect any possible hardware problems and undertake the appropriate measures.

See also:

[Backup Verify \[Page 118\]](#)



-V|-VERSION

This BRBACKUP command option displays detailed information on the program modules and patches.

Input syntax: -v

25.2.6 BRBACKUP Logs

For more information, see:

- [Names of the BRBACKUP Detail Logs \[Page 377\]](#)
- [BRBACKUP Detail Log \[Page 378\]](#)
- [Log Supplements \[Page 484\]](#)
- [BRBACKUP Summary Log \[Page 379\]](#)



Names of the BRBACKUP Detail Logs

Every detail log contains a name with the following format:

b<encoded timestamp>.xyz

The first characters indicate the encoded time the backup was performed (action ID). The extension (function ID) indicates the type of backup.

Possible values for x:

- **a**: Whole database was backed up (`backup_mode = all|all_data`).
- **p**: One or more tablespaces or files were backed up (that is, partial backup).
- **f**: Full (level 0) database backup (`backup_mode = full`)
- **i**: Incremental (level 1) database backup (`back_mode = incr`)

Possible values for y:

- **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|offline_stop`).

Possible values for z (specification of backup devices):

- **t**: Tape device (`backup_dev_type = tape|tape_auto|tape_box`).
- **d**: Local 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`).
- **s**: Remote disk (`backup_dev_type = stage|stage_copy|stage_standby`)

Other function IDs:

25.2 BRBACKUP

- **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.
- **dbv**: The BRBACKUP option `-w only_dbv` was used to verify the internal database block structure with `DBVERIFY`, but no backup was started.
- **rmp**: The BRBACKUP option `-d rman_prep` was used to prepare for backup with the Oracle Recovery Manager (RMAN), but no backup was started.
- **ddb**: The BRBACKUP option `-db` was used to delete a disk backup, but no backup was started.



BRBACKUP Detail Log

This section describes the information contained in a detail log (`a<encoded timestamp>.<ext>`), see [Names of the BRBACKUP Detail Logs \[Page 377\]](#).

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 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 \[Page 484\]](#), [-o|-output \[Page 372\]](#) and [-w|-verify \[Page 376\]](#).

This information indicates which database file, non-database file, or directory, or which profile and log were saved and 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 BRRESTORE), 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 messages.



BRBACKUP 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
- Total compression rate (appears only when compression was used)
- Number of files backed up:
 - `all [*]`: all database files

25.2 BRBACKUP

- `all_data[*]`: all pure data tablespaces
 - `partial[*]`: partial backup
 - `non_db`: only non-database files (for example, `sap_dir`, `ora_dir`)
- "*" means that non-database files were also backed up.
- Value of the parameter [backup_type \[Page 450\]](#)
 - Value of the parameter [backup_dev_type \[Page 447\]](#)
 - Internal flags for the BRBACKUP command options
 - BRBACKUP version

25.2.7 Profile Parameters and BRBACKUP Command Options

Save-Set Grouping and Preparation Run

To define the size of the save sets by the number of files they contain, make the following entry in the [initialization profile init<DBSID>.sap \[Page 445\]](#):

```
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 save sets as follows:

- `brbackup -d rman_prep`
- SAPDBA menu: *Backup database* → *Backup function*

For more information, see [RMAN Save-Set Grouping \[Page 151\]](#).

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 for backup to:

- Local tape devices
 - `backup_dev_type = tape|tape_auto|tape_box`
 - `tape_copy_cmd = rman|rman_dd`
- Remote tape devices
 - `backup_dev_type = pipe|pipe_auto|pipe_box`
 - `tape_copy_cmd = rman|rman_dd`
- Remote disk (incremental)
 - `backup_mode = incr`
 - `backup_dev_type = stage`
 - `remote_user = "<user name> [<password>]"` (<password> is required for SAPFTP)
 - `remote_host = <host name>`

For more information, see [RMAN Backup with the SAP Backup Library \[Page 142\]](#).

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 save sets are formed and the parallel backup to different media.

- `rman_channels`
Number of channels allocated to the parallel processes
- `rman_filesperset`
Number of files in each save set
Default is 0 (that is, one file in each save set for normal backups, or all files in one save set for incremental backups)
- `rman_kbytes`
- `rman_readrate`
- `rman_maxopenfile`
- `rman_setsize`
- `rman_diskratio`
- `rman_pool`
- `rman_duplex`
- `rman_proxy`
- `rman_parms`

For more information on these parameters, see [Initialization Profile init<DBSID>.sap \[Page 445\]](#) and the Oracle backup and recovery documentation.

For more information, see [RMAN Backup with an External Backup Library \[Page 144\]](#).

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:

25.2 BRBACKUP

```
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
```

For more information, see [RMAN Incremental Backups Without a Backup Library \[Page 146\]](#).

Backup of Offline Redo Log Files with RMAN

If you are using the SAP backup library, set the following entries in the initialization file `init<DBSID>.sap` or use the corresponding command options:

- `backup_dev_type = tape|pipe`
- `tape_copy_cmd = rman|rman_dd`

If you are using an external backup library, use the following setting:

```
backup_dev_type = rman_util
```

For backups to disk with RMAN but without a backup library, use the following settings:

- `backup_dev_type = disk`
- `disk_copy_cmd = rman`

Then start BRARCHIVE, for example as follows:

```
brarchive -sd -c
```

For more information, see [RMAN Backup of the Offline Redo Log File \[Page 149\]](#).

Restoring Incremental Backups

To make the database consistent again, for example after a media error, you can use the following BRRESTORE command (RMAN redoes the changes in the files that have been affected):

1. If necessary, restore of the control file and the online redo log files from the last incremental backup:

```
brrestore -b last -m 0[,00]
```

2. Restore affected files of a full backup (level 0):

```
brrestore -b <brb_log_name> -m all|<object list>|..
```

3. Restore of the last incremental backup (level 1):

```
brrestore -b last -m incr
```

4. Applying the offline redo log files with Oracle SQLPLUS.

For more information, see [RMAN Restore of Incremental Backups \[Page 141\]](#).

Restoring Incremental Backups with Structural Changes

In an incremental backup with structural changes, the new files are backed up in full to a second save set in subsequent incremental backups. The following save sets 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 save set allows a precise specification of which files are to be restored.

Restore of changes to all files that were in the database at the time of the last full backup at level 0 (first save set of the last incremental backup):

```
brrestore -b last -m incr_only
```

Restore of the files that have been added since the last full backup at level 0 (second save set of the last incremental backup):

```
brrestore -b last -m incr_full
```

Restore of the whole incremental backup (both save sets, if they exist):

```
brrestore -b last -m incr
```

For more information, see [RMAN Incremental Backups After Structural Changes \[Page 140\]](#).

See also:

Oracle documentation

25.3 BRARCHIVE

Use

This SAP tool for the Oracle database enables you to archive offline redo log files. For further information about features common to both tools, see [Common Features of BRBACKUP and BRARCHIVE \[Page 357\]](#).

Prerequisites

Make sure that the [initialization profile init<DBSID>.sap \[Page 445\]](#) is configured properly and use the appropriate [command options for BRARCHIVE \[Page 384\]](#).

Features

- 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 \[Page 360\]](#), 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 \[Page 390\]](#).
- Use the option [-f|-fill \[Page 387\]](#) to archive the offline redo log files permanently. In this way, you can make sure that the archiving directory does not fill up. See.
- 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.

See also:

- [Hardware Compression for BRARCHIVE \[Page 383\]](#)
- [Logging \[Page 384\]](#)

25.3.1 Hardware Compression for BRARCHIVE

25.3 BRARCHIVE

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 parameter, [tape_size_arch](#) [Page 479], for BRARCHIVE. If you do this, changes to individual parameters do **not** affect the next database backup with BRBACKUP.

See also:

[Tape Devices with Hardware Compression](#) [Page 101]

25.3.2 Logging for BRARCHIVE

BRARCHIVE writes the following file system logs:

- `<SAPDATA_HOME>/saparch/a<encoded timestamp>.<ext>`

Every detail log contains information about the actions performed during an archiving run.

- `<SAPDATA_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 392].

25.3.3 Command Options for BRARCHIVE

This section describes the command options for [BRARCHIVE](#) [Page 383]. If you start BRBACKUP without command options, the values in the [initialization profile](#) `init<DBSID>.sap` [Page 445] are used. Unless otherwise specified in the initialization profile `init<DBSID>`, BRARCHIVE starts archiving all the offline redo log files to tape, with a storage capacity of 1200 MB and no file compression.

For more information about features common to both tools, see [Common Features of BRBACKUP and BRARCHIVE](#) [Page 357].

If you use BRARCHIVE with command options, these override the corresponding values in the initialization profile. To use the options, you can specify either the letter indicated or the complete word.

```
brarchive
[-a|-archive
[-b|-backup [<brb_options>]]
[-c|-confirm [<force>]]
[-d|-device
tape|disk|pipe|tape_auto|pipe_auto|tape_box|pipe_box|util_file|stage
|rman_util]
[-f|-fill [<number>|stop|suspend|resume]]
```

```

[-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|-cde|-
copy_delete_save]
[-u|-user [<user>[/<password>]]]
[-v|-volume <volume>|<volume list>|SCRATCH]
[-w|-verify]
[-V|-VERSION]

```

See also:

[-a|-archive \[Page 385\]](#)
[-b|-backup \[Page 386\]](#)
[-c|-confirm \[Page 386\]](#)
[-d|-device \[Page 387\]](#)
[-f|-fill \[Page 387\]](#)
[-h|-help \[Page 388\]](#)
[-i|-initialize \[Page 388\]](#)
[-k|-compress \[Page 388\]](#)
[-l|-language \[Page 388\]](#)
[-m|-modify \[Page 389\]](#)
[-n|-number \[Page 389\]](#)
[-o|-output \[Page 389\]](#)
[-p|-profile \[Page 389\]](#)
[-q|-query \[Page 389\]](#)
[-r|-parfile \[Page 390\]](#)
[-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cde \[Page 390\]](#)
[-u|-user \[Page 391\]](#)
[-v|-volume \[Page 391\]](#)
[-w|-verify \[Page 391\]](#)

**-a|-archive**

This BRARCHIVE command option backs up offline redo log files saved earlier on disk.

Input syntax: -a

25.3 BRARCHIVE

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.

The option `-a|-archive` can only be used in combination with the BRARCHIVE options `-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 \[Page 109\]](#)

**-b|-backup**

This BRARCHIVE command option starts BRBACKUP at the end of BRARCHIVE processing.

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 \[Page 363\]](#)



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 \[Page 364\]](#)).



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 also:

[BRBACKUP and BRARCHIVE Backups in One Run \[Page 116\]](#)

[Unattended Backup \[Page 114\]](#)

**-c|-confirm**

This BRARCHIVE command option backs up in attended mode.

Input syntax: `-c [force]`

Default value: Confirmation messages are issued and user entry is expected.

See [-c|-confirm \[Page 366\]](#).



-d|-device

This BRARCHIVE command option defines the backup device type.

Input syntax:

`-d`
`tape|disk|pipe|tape_auto|pipe_auto|tape_box|pipe_box|util_file|stage`
`|rman_util`

Default value: `tape`

The following backup media are supported:

- `tape`: Local tape device.
- `disk`: Local disk. Normally only use the option for archiving to disk in exceptional cases, such as two-step archiving. See [archive_copy_dir \[Page 445\]](#).
- `stage`: Remote disk.

Offline redo log files should always be archived finally to tape, 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 Automatic Tape Changers \[Page 176\]](#).
- `util_file`: Archiving with BACKINT – see the corresponding option in [-d|-device \[Page 367\]](#).
- `rman_util`: Archiving with Oracle Recover Manager (RMAN) and an external backup tool. See [RMAN Backup with an External Backup Library \[Page 144\]](#).

See also:

Parameters in `init<DBSID>.sap`: [backup_dev_type \[Page 447\]](#)



-f|-fill

This BRARCHIVE command option fills up a backup volume by waiting for the next offline redo log files.

Input syntax: `-f [<number>|stop|suspend]|resume]`

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.

25.3 BRARCHIVE

- **<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 \[Page 320\]](#).
- **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 BRARCHIVE Detail Logs \[Page 392\]](#).
- **suspend**: Suspend BRARCHIVE processing that was started with `-f`. This can be useful if, for example, you want to perform an offline backup and later restart BRARCHIVE.
- **resume**: Resume processing suspended with `-f suspend`.

**-h|-help**

This BRARCHIVE command option provides help information.

Input syntax: `-h [version]`

Default value: No help

See [-h|-help \[Page 369\]](#).

**-i|-initialize**

This BRARCHIVE command option initializes tape volumes.

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 \[Page 91\]](#).

If you specify the option `-i` to initialize volumes (and `-v` has not been specified), BRARCHIVE initializes the volumes specified in `volume_archive`.

See [-i|-initialize \[Page 369\]](#).

**-k|-compress**

This BRARCHIVE command option sets compression mode.

Input syntax: `-k no|yes|hardware|only`

Default value: `no` (no compression)

See [-k|-compress \[Page 370\]](#).

The option, `-k only`, has practically no meaning for BRARCHIVE as the information is not saved in the database. See [Hardware Compression for BRBACKUP \[Page 363\]](#)

See [-k|-compress \[Page 370\]](#)

**-l|-language**

This BRARCHIVE command option sets message language.

Input syntax: `-l E|D`

Default value: `E`

See [-l|-language \[Page 370\]](#).



-m|-modify

This BRARCHIVE command option applies the offline redo log files to a standby database.

Input syntax: **-m** [**<delay>**]

Default value: No delay

Delay: The offline redo log files that are created are sent to the standby database before they are processed. There they can be applied 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 apply the offline redo log files.

For more information, see: [Standby Database \[Page 162\]](#).



-n|-number

This BRARCHIVE command option defines the number of offline redo log files or tape volumes to be processed.

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`.



-o|-output

This BRARCHIVE command option prints extra information to the log file.

Input syntax: **-o** **dist** | **time** [, **time** | **dist**]

Default value: The BRARCHIVE detail log is written in normal form. See [BRARCHIVE Detail Log \[Page 393\]](#).

See [-o|-output \[Page 372\]](#).



-p|-profile

This BRARCHIVE command option defines the profile name.

Input syntax: **-p** **<profile>**

Default value: `init<DBSID>.sap`

See [-p|-profile \[Page 372\]](#).



-q|-query

25.3 BRARCHIVE

This BRARCHIVE command option sets the query mode.

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 \[Page 373\]](#).



-r|-parfile

This BRARCHIVE command option defines the BACKINT parameter file.

Input syntax: `-r <parameter_file>`

Default value: No parameter file

See [-r|-parfile \[Page 373\]](#).



-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cds

This BRARCHIVE command option defines the BRARCHIVE function to be performed.

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|-cds|-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.
- **-cds | -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 **-cds** 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 \[Page 446\]](#).



-u|-user

This BRARCHIVE command option defines user name and password.

Input syntax: **-u** [**<user>** [**/<password>**]]

Default value: **system/manager**

See [-u|-user \[Page 375\]](#).



-v|-volume

This BRARCHIVE command option defines the tape volumes to be used.

Input syntax: **-v** **<volume>** | **<volume list>** | **SCRATCH**

Default value: defined in `init<DBSID>.sap`

If the option **-v** is not used, BRARCHIVE uses the volumes (tapes) specified in parameter `volume_archive`. For more information, see [-v|-volume \[Page 375\]](#).



-w|-verify

This BRARCHIVE command option verifies the backup after the offline redo log files were saved.

Input syntax: **-w**

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, 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.

25.3 BRARCHIVE

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 `-cds` is used, the sizes of the archived files are determined and checked.

See [Log Supplements \[Page 484\]](#).

Note that verification approximately doubles 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.



-V|-VERSION

This BRARCHIVE command option displays detailed information on the program modules and patches.

Input syntax: `-v`

25.3.4 BRARCHIVE Logs

For more information, see:

- [Names of the BRARCHIVE Detail Logs \[Page 392\]](#)
- [BRARCHIVE Detail Log \[Page 393\]](#)
- [BRARCHIVE Summary Log \[Page 394\]](#)



Names of the BRARCHIVE 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 - cds).
- **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 offline redo log files.
- **fst:** The BRARCHIVE option -f stop was used to stop a BRARCHIVE run.



BRARCHIVE Detail Log

This section describes the information contained in a detail log (a<encoded timestamp>.<ext>, see [Names of the BRARCHIVE Detail Logs \[Page 392\]](#)).

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 for BRBACKUP and BRARCHIVE \[Page 484\]](#), [-o|-output \[Page 389\]](#) and [-w|-verify \[Page 391\]](#).

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 messages (for example, the total number of processed offline redo log files).

Log Supplements

25.3 BRARCHIVE

Using the option `-o dist|time [,time|dist]` and/or the option `-w` causes the detail log to be supplemented.

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 for BRBACKUP and BRARCHIVE \[Page 484\]](#).



BRARCHIVE 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, starting with the following:
 - `#` for `tape_copy_cmd = cpio`
 - `$` for `tape_copy_cmd = dd`
 - `+` for `tape_copy_cmd = rman`
 - `-` for `tape_copy_cmd = rman_dd`
 - `*` for `backup_dev_type = util_file`
 - `&` for `backup_dev_type = rman_util`
 - `@` for backup from disk backup with BACKINT
 - 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, the total compression rate (if compressed), internal flag for the BRARCHIVE command options, and BRARCHIVE version.

- **#DISK/#STAGE** This information appears if you used the BRARCHIVE feature for archiving to local or remote 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/#STAGESAV** Archiving to disk: action ID, function ID, name of the file on the local or remote disk, timestamp, compression rate and the size of the compressed file (if it was compressed).
- **#DISKDEL/#STAGEDDEL** Archiving to disk: information (action ID, function ID, timestamp), when the file was deleted from archive directory.
- **#DELDISK** A copy of an offline redo log file on disk was deleted: information (action ID, function ID, timestamp) when the file was deleted from the disk backup directory
- **#APPLIED** Offline redo log file was applied to the standby database.
- **##** You can enter user comments into the BRARCHIVE summary log with this prefix.

25.4 BRRESTORE

Use

This SAP tool enables you to restore an entire database backup or parts of it, when the backup was performed with [BRBACKUP \[Page 360\]](#). 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 \[Page 383\]](#). This operation can be performed at the same time as the restore of the corresponding backup.

- BRRESTORE can run unattended when option `-c force` is set. 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.

Integration

BRRESTORE only restores the selected backup. It does **not** recover the database. To do this, start the recovery afterwards using one of the following:

- The SAPDBA tool – see [SAPDBA Database Recovery \[Page 321\]](#)
- The SQLPLUS tool from Oracle – see [Recovery with SQLPLUS \[Page 487\]](#)

For more information, see:

- [Restoring Files \[Page 396\]](#)
- [Examples of BRRESTORE Runs \[Page 397\]](#)

25.4 BRRESTORE

25.4.1 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 \[Page 398\]](#).

BRRESTORE also requires several parameters to be configured in the [initialization profile init<DBSID>.sap \[Page 445\]](#).

- The options `-d`, `-k`, `-m`, and `-r` can be preset using the appropriate BRRESTORE profile parameters. See [Effects of the Command Options \[Page 359\]](#).
- Only one of the options `-a`, `-b`, or `-n` can be set. If you do not select any of these options, `-b last` is selected.

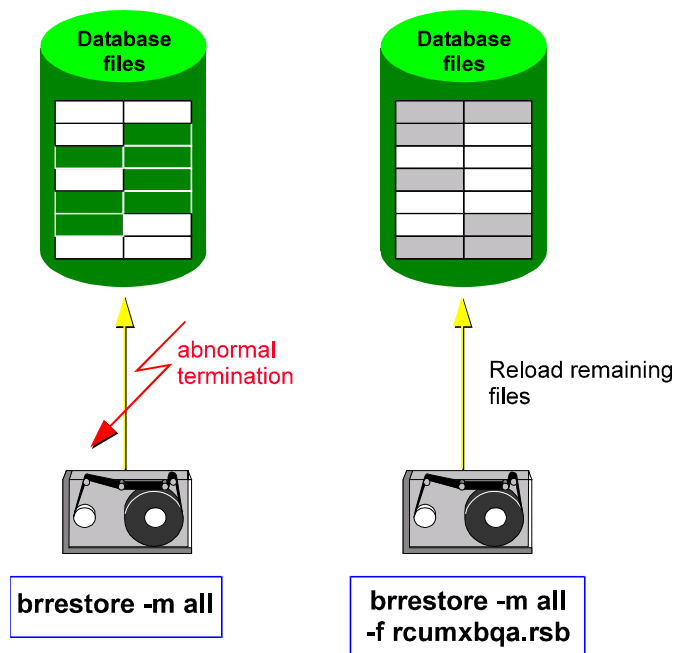
However, BRRESTORE can be started with option `-a` in parallel to BRRESTORE with option `-b` or `-n`.

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 \[Page 459\]](#).

- 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>` (for example, 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 `SAPDATA_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 offline redo log files can be addressed using their log sequence numbers.

25.4.2 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 402] and specify the log name of the BRRESTORE run to be completed. If you want to complete several runs, specify all log names individually.

You can also use option `-f last` to complete the last BRRESTORE run, or option `-f <days>` to complete all BRRESTORE runs started in the last `<day>` days. For more information, see [fillup](#) [Page 402].

See also:

[Completion of BRBACKUP Backups](#) [Page 362]

25.4.3 Examples of BRRESTORE Runs

```
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 SAPDBA executable.

```
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.

25.4 BRRESTORE

```
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 working directory.

25.4.4 Command Options for BRRESTORE

This section describes the command options for [BRRESTORE \[Page 395\]](#). If you start BRRESTORE without command options, the values in the [Initialization Profile `init<DBSID>.sap` \[Page 445\]](#) are used. Unless otherwise specified in the initialization profile `init<DBSID>.sap`, BRRESTORE restores the files of all tablespaces from the last successful backup of the database.

If you use BRRESTORE with command options (see below), these override the corresponding values in the initialization profile. To use the options, you can specify either the 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 | -b1 | -backup1 <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 | stage | rman_util]
- e | -execute <number>]
- f | -fillup <log_name1>[,<log_name2>,...] | <no. of days> | last]
- h | -help [version]]
- k | -compress no | yes | hardware]
- l | -language E | D]
- m | -mode
all | all_data | full | incr | incr_only | incr_full | <tablespace>[=<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 | init_all | all_
log[=<rest_dir>]]
- o | -output dist | time[,time | dist]]
```

```

[-p|-profile <profile>]
[-q|-query [check]]
[-r|-parfile <parameter_file>]
[-w|-verify [use_dbv]]
[-V|-VERSION]

```

See also:

[-a|-archive|-a1|-archive1 \[Page 399\]](#)

[-a2|-archive2 \[Page 400\]](#)

[-b|-backup|-b1|-backup1 \[Page 400\]](#)

[-b2|-backup2 \[Page 400\]](#)

[-c|-confirm \[Page 401\]](#)

[-d|-device \[Page 401\]](#)

[-e|-execute \[Page 402\]](#)

[-h|-help \[Page 402\]](#)

[-k|-compress \[Page 402\]](#)

[-l|-language \[Page 403\]](#)

[-m|-mode \[Page 403\]](#)

[-n|-number \[Page 404\]](#)

[-o|-output \[Page 404\]](#)

[-p|-profile \[Page 405\]](#)

[-q|-query \[Page 405\]](#)

[-r|-parfile \[Page 405\]](#)

[-w|-verify \[Page 405\]](#)

[-V|-VERSION \[Page 406\]](#)

-a|-archive|-a1|-archive1

This BRRESTORE command option restores offline redo log files from the first copy.

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>`: Database instance ID. Must be specified only for Oracle Parallel Server (OPS).
- `<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.

25.4 BRRESTORE

- **<rest_dir>**: The restore directory where the archived redo log files will be restored. If you do not specify a directory, the archiving directory (<SAPDATA_HOME>/saparch) is selected.
- **<log_no_list>**: You can combine any specifications for the log sequence intervals. Separate the individual names with commas. Do not use blanks.

Under certain circumstances, you can restore archived redo log files from several volumes simultaneously, in parallel. See [Restoring Files \[Page 396\]](#).

-a2|-archive2

This BRRESTORE command option restores offline redo log files from the second copy.

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>**: Database instance ID. Must be specified only for Oracle Parallel Server (OPS).
- **<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 \[Page 399\]](#).



-b|-backup|b1|backup1

This BRRESTORE command option restores database files saved by BRBACKUP.

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

This BRRESTORE command option restores files calling backup tools with the BACKINT interface.

Input syntax: `-b2 <util_backup_id>|#NULL`

<util_backup_id>: Backup ID of backup with an external tool

#NULL: Restore from the last BACKINT backup.

With this option you can reload backups which were executed with an external backup tool via the interface BACKINT. Use option -m to define which files should be restored.

See also:

[External Backup Programs \[Page 180\]](#)



-c|-confirm

This BRRESTORE command option restores in unattended mode.

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 \[Page 111\]](#)



Do not use the option **-c force** when recovering a database. Follow the BRRESTORE confirmation messages in this case.



-d|-device

This BRRESTORE command option defines the restore device type.

Input syntax:

-d *tape* | *disk* | *pipe* | *tape_auto* | *pipe_auto* | *tape_box* | *pipe_box* | *util_file* | *stage* | *rman_util*

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:** Local disk.
- **stage:** Remote disk.
- **tape:** Local tape device.
- **pipe:** 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**.
- **rman_util :** Restoring with Oracle Recover Manager (RMAN) and an external backup tool. See [RMAN Backup with an External Backup Library \[Page 144\]](#).

25.4 BRRESTORE

See also:

Parameters in `init<DBSID>.sap`: [backup_dev_type \[Page 447\]](#).

**-e|-execute**

This BRRESTORE command option executes the restore in parallel.

Input syntax: `-e <n>`

Default value: 0

See [-e|-execute \[Page 368\]](#).

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**

This BRRESTORE command option completes the restore run.

Input syntax: `-f <log_name1>[,<log_name2>,...] | <no. of days lost> | last`

Default value: `last`

BRRESTORE determines which files still have to be restored using the successfully restored files in one of the following:

- One or more BRRESTORE logs `<log_name>`
- All restores in a defined number of preceding days `<no. of days>`
- The last restore run

See also:

[Completion of BRRESTORE Runs \[Page 396\]](#)

**-h|-help**

This BRRESTORE command option provides help information.

Input syntax: `-h [version]`

Default value: No help

See [-h|-help \[Page 369\]](#).

**-k|-compress**

This BRRESTORE command option sets compression mode.

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 \[Page 454\]](#).



-l|-language

This BRRESTORE command option sets the message language.

Input syntax: `-l E|D`

Default value: `E`

See [-l|-language \[Page 370\]](#).



-m|-mode

This BRRESTORE command option defines the files to be restored.

Input syntax:

```
-m
all|all_data|full|incr|incr_only|incr_full|<tablespace>[=<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 recreated.
- **incr:** Restore an incremental backup with Oracle Recovery Manager (RMAN). See [Restoring Incremental Backups \[Page 330\]](#).
- **incr_only:** Restore changes to all files that were in the database at the time of the last full backup. See "Restoring Incremental Backups with Structural Changes" in [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).
- **incr_full:** Restore files that have been added to the database since the last full backup. See "Restoring Incremental Backups with Structural Changes" in [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).
- **<tablespace>:** 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 `SAPDATA_HOME` directory and an additional generic specification (for example, `sapdata<n>`) in the path.

25.4 BRRESTORE

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 \[Page 461\]](#)
- **<object_list>**: You can specify a list of tablespaces or files, or combine the key words `all` 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 \[Page 449\]](#).

-n|-number

This BRRESTORE command option enables you to restore a file directly from the volume without having to specify the name of the backup log or the log sequence number.

Input syntax:

```
-n
<file_pos>|init_ora|init_sap|reorg_log|det_log|sum_log|init_all|all_
log[=<rest_dir>]
```

Default: None.

- **<file_pos>**: Specify the position of the file on the tape.
- **init_ora**: The profiles `init<DBSID>.ora` and `init<DBSID>.dba` 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.
- **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.
- **init_all**: Restores `init<DBSID>.ora`, `init<DBSID>.dba` and `init<DBSID>.sap` in one run.
- **all_log**: Restores `reorg<DBSID>.log`, `struct<DBSID>.log`, detail and summary BRARCHIVE or BRBACKUP logs in one run.
- **<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

This BRRESTORE command option prints extra information to the log file.

Input syntax: `-o dist|time[,time|dist]`

Default value: The BRRESTORE detail log is written normally. See [BRRESTORE Detail Log \[Page 406\]](#).

See [-o|-output \[Page 372\]](#).



-p|-profile

This BRRESTORE command option defines the profile name.

Input syntax: `-p <profile>`

Default value: `init<DBSID>.sap`

See [-p|-profile \[Page 372\]](#).



-q|-query

This BRRESTORE command option sets the query mode.

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

This BRRESTORE command option defines the BACKINT parameter file.

Input syntax: `-r <parameter_file>`

Default value: No parameter file

See [-r|-parfile \[Page 373\]](#).



-w|-verify

This BRRESTORE command option verifies a backup of database files (BRBACKUP) or offline redo log files (BRARCHIVE).

Input syntax: `-w [use_dbv]`

Default value: No verification

Without `use_dbv`, files are read from backup media but **not** stored on disk. With `use_dbv`, files are restored to [compress_dir \[Page 455\]](#), verified with DBVERIFY and then deleted. In either case, a normal restore is **not** performed.



Restoring the last check, followed by a check of the Oracle block structure using DBVERIFY:

```
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
```

25.4 BRRESTORE

See also:

[Backup Verify \[Page 118\]](#)



-V|-VERSION

This BRRESTORE command option displays detailed information on the program modules and patches.

Input syntax: -v

25.4.5 BRRESTORE Logs

For more information, see:

- [Names of the BRRESTORE Detail Logs \[Page 406\]](#)
- [BRRESTORE Detail Log \[Page 406\]](#)
- [BRRESTORE Summary Log \[Page 407\]](#)



Names of the BRRESTORE 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 restore.

Possible function IDs:

- **rsb**: Restore from a BRBACKUP backup (option -b | -backup | -b2).
- **rsa**: Restore the offline redo log files specified by the log sequence numbers (option -a | -archive | -a1 | -a2).
- **rsf**: Restore a file characterized by its position on the backup volume (option -n | -number).
- **qur**: The BRRESTORE option -q or -q check was used to display which volumes are to be used for restore or to make sure that those volumes were actually mounted. No restore was started.



BRRESTORE 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 as it was saved on the tape
 - 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

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 \[Page 484\]](#) and [_o|-output \[Page 404\]](#). The details of the restore times (`-o time`) refer to the restore process.



BRRESTORE Summary Log

You can display a brief entry for each restore in the summary log `rest<DBSID>.log`. The entries in the file provide the following information about each restore using BRRESTORE:

- Action ID (encoded timestamp of the log name)
- Function ID (extension of the log name)
- Timestamp (date, time) specifying the start of the restore
- Timestamp (date, time) specifying the end of the restore
- Return code
- Value of [restore_mode \[Page 464\]](#)
- Value of [backup_dev_type \[Page 447\]](#)
- Internal flags for the BRRESTORE command options
- BRRESTORE version

25.5 BRCONNECT

Use

The SAP tool BRCONNECT for Oracle databases is used as:

- A database administration tool, which you can call yourself from the command line
- A utility tool, which [BRBACKUP \[Page 360\]](#) calls in the background

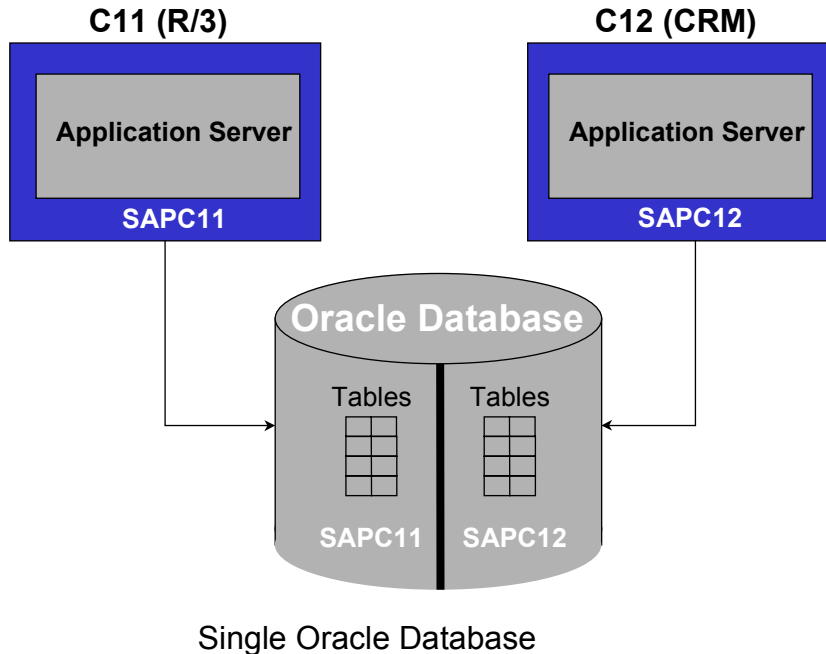


BRCONNECT was originally only a utility tool. Now you can also start it from the command line and it has taken over some of the functionality of SAPDBA. Therefore, certain SAPDBA functions are no longer being developed.

25.5 BRCONNECT

We **strongly recommend** you to use BRCONNECT rather than SAPDBA where the two tools perform the same tasks.

BRCONNECT is specially designed to administer multi-schema databases, in which you have more than one SAP System in the same Oracle database:



Features

- Database administration tool

As a database administration tool, BRCONNECT has the following [functions \[Page 430\]](#):

- Main functions, which write a [detailed \[Page 444\]](#) and a [summary \[Page 444\]](#) log
 - [Check database system \[Page 409\]](#), [-f check \[Page 431\]](#)
 - [Adapt next extents \[Page 419\]](#), [-f next \[Page 435\]](#)
 - [Update statistics \[Page 420\]](#), [-f stats \[Page 436\]](#)
 - [Clean up old logs and traces \[Page 427\]](#), [-f cleanup \[Page 432\]](#)
- [Additional functions \[Page 428\]](#), which only write messages to the standard output device
 - Change passwords of SAP database users, [-f chpass \[Page 432\]](#)
 - Create global synonyms, [-f crsyn \[Page 434\]](#) – these are used by BR tools and SAPDBA
 - Start database, [-f dbstart \[Page 430\]](#)
 - Stop database, [-f dbshut \[Page 434\]](#)
 - Determine database state, [-f dbstate \[Page 430\]](#)
- Utility tool - monitor database status during a backup

BRBACKUP starts BRCONNECT during the backup to see if the status of the database corresponds to the backup mode. 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 the state of the database changes unexpectedly during the backup, BRCONNECT terminates the backup and displays the messages 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.

BRCONNECT has many parameters, which you can specify in the [Initialization Profile init<DBSID>.sap](#) [Page 445].

Activities

For more information about using BRCONNECT from the command line, see [Command Options for BRCONNECT](#) [Page 429].

25.5.1 Database System Check with BRCONNECT

Use

You can use [BRCONNECT](#) [Page 407] to check the Oracle database system. The aim is to prevent database problems that might lead to downtime for the database.



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -check](#) [Page 347]. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

You can use BRCONNECT to check the following conditions:

- Database administration, such as configuration, space management, state of the database, and so on
- Database operations, such as backup results, failed operations, and so on
- Critical database messages in the Oracle alert file, such as ORA-00600
- Incorrectly set database profile parameters in the `init<DBSID>.ora` file, such as `log_archive_start = false`

When a critical situation is discovered, BRCONNECT writes an alert message to the [detail log](#) [Page 444] and to the results table DBMSGORA.

Integration

The check conditions are specified in the control table DBCHECKORA. You can change these with transaction DB17. Detected alerts are also reported to the database monitor (transaction RZ20). For more information, see [Monitoring the Oracle Database](#) [Ext.].

Prerequisites

BRCONNECT can also use internal default conditions, which mostly correspond to the initial state of the DBCHECKORA table when we deliver it.

25.5 BRCONNECT



For up-to-date information on the BRCONNECT default conditions, see SAP Note 435290.

BRCONNECT Default Conditions for Database Administration

See [BRCONNECT Default Conditions for Database Administration \[Page 411\]](#).

BRCONNECT Default Conditions for Database Operations

See [BRCONNECT Default Conditions for Database Operations \[Page 416\]](#).

BRCONNECT Default Conditions for Database Messages

You can enter any Oracle error codes or error text as a condition name for this condition type. BRCONNECT searches the Oracle Alert log for corresponding Oracle error messages and might generate Alert messages. The following Oracle error codes are taken into account for the standard test conditions (-d option):

```
ORA-00272 Error writing archive log
ORA-00376 Database file cannot be read
ORA-00600 Oracle internal error
ORA-01113 Data file needs media recovery
ORA-01115 I/O error reading database file
ORA-01122 File verification check failed
ORA-01135 Database file is offline
ORA-01149 Cannot shutdown - data file in backup mode
ORA-01555 Snapshot too old
ORA-01562 Failed to extend rollback segment
ORA-01578 Database block corrupted
ORA-03113 End-of-file on communication channel
ORA-07445 Exception encountered: core dump
Checkpoint not complete
```



You can easily add any Oracle error codes as new test conditions for database messages using transaction DB17. You can also search any texts (character strings) in the Oracle Alert file by specifying the text in the `PARAM` field. Since the search in the Oracle Alert file takes the upper/lower case spelling into account and the entry in the `PARAM` field is always converted into upper case letters using DB17, the entry must be made using SQL with SQLPLUS, as in the following example:



```
INSERT INTO DBCHECKORA (TYPE, PARAM, OBJECT, ACTIVE,
SEVERITY, CHKOP, CHKVAL, UNIT, CHKREP, REPUNIT, MODFLAG,
MODDATE, MODUSER, REACTION, CORRTYPE, CORRNAME, CHKDESC)
VALUES ('ORA', 'Checkpoint not complete', ' ', 'Y', 'W',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'D',
'Increase the size of online redo log files',
```

```
'Cannot switch to the next online redo log due to pending  
checkpoint');
```

BRCONNECT Default Conditions for Database Profile Parameters

These test conditions check the values of Oracle parameters. The standard test conditions for the database profile (-d option) correspond to the current SAP recommendations described in note 124361 and 180605 (SAP BW).



You can use transaction DB17 to easily adjust the test conditions for the database profile parameters, depending on the changed recommendations and for new Oracle releases.

Downward compatibility

For earlier SAP releases (up to Release 4.0B), BRCONNECT for SAP Web Application Server 6.10 supports the old SAPDBA test conditions. In Releases 4.6X, however, you can replace the old test conditions in table DBCHECKORA with the new ones. To do this, run the following SQL script:

```
sapserv*:/general/misc/sapdba/scripts/dbcheckora.sql  
SQLPLUS> connect sapr3/<pwd>  
SQLPLUS> @dbcheckora
```

Then you can maintain the check conditions as of the specific service packages using transaction DB17 (see note 427673 for this). If required, you can also maintain the check conditions using native SQL.

Activities

- You run the checks regularly (for example, daily). We recommend you to use the Database Planning Calendar in the SAP System for this. For more information, see [Database System Check \[Ext.\]](#).
- You use transaction DB16 to view alerts written by BRCONNECT to the results table DBMSGORA. For more information, see [Displaying Alert Messages from Database System Check \[Ext.\]](#).
- You use transaction DB17 to configure database system check. This includes activating or deactivating check conditions and changing the threshold and severity levels (that is, error, warning, or exception). For more information, see [Configuring Database System Check \(Oracle\) \[Ext.\]](#).
- You can exclude specified tables or indexes from the checks using the `check_exclude` parameter.

For more information on the command line options for the database checks, see [-f check \[Page 431\]](#).



BRCONNECT Default Conditions for Database Administration

The check conditions for database administration are specified in the control table DBCHECKORA. For more information, see [Database System Check with BRCONNECT \[Page 409\]](#).

25.5 BRCONNECT

Condition	Severity Level	Description
NOARCHIVELOG_MODE (formerly NOARCHIVELOG)	Error	<p>Checks whether the database is in NOARCHIVELOG mode, which is not allowed for production databases.</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.</p>
ARCHIVER_STUCK (formerly ARCHIVE_STUCK)	Warning	<p>Checks the highest fill level of the archiving directory (by default, saparch).</p> <p>The OBJECT field is not specified for this condition.</p> <p>Possible test operands, threshold values, value units:</p> <ul style="list-style-type: none"> • >, >= <number> The space used is larger than (or the same as) <number> % of the total space in the archiving directory. • >, >= <number> The space used is larger than (or the same as) <number> K M G bytes. • <, <= <number> The free space is smaller than (or the same as) <number> % of the total space in the archiving directory. • <, <= <number> The free space is smaller than (or the same as) <number> K M G bytes.
FILE_SYSTEM_FULL (formerly FS_FULL)	Warning	<p>This condition checks the fill level of file systems on the database host. All file systems are checked by default against the same threshold value. However, you can define different threshold values for individual file systems by specifying the file system in the OBJECT field of the DBCHECKORA table (transaction DB17). Here you can use the following keywords for the database file systems:</p> <p>ORACLE_HOME, SAPDATA_HOME, SAPDATA1, SAPDATA2, . . . , SAPARCH, ORIGLOG, MIRRLOG, SAPBACKUP, SAPCHECK, SAPREORG and SAPTRACE.</p> <p>Alternatively, you can specify any other file systems by entering the full path of a directory from this file system into the field (you can use upper/lower case spelling in DB17 only as of the Service Packages mentioned in note 427673, otherwise you have to use native SQL).</p> <p>Possible test operands, threshold values, value units:</p> <ul style="list-style-type: none"> • >, >= <number>

		<p>The space used is larger than (or the same as) <number> % of the total space in the file system.</p> <ul style="list-style-type: none"> • >, >= <number> <p>The space used is larger than (or the same as) <number> K M G bytes.</p> <ul style="list-style-type: none"> • <, <= <number> <p>The free space is less than (or the same as) <number> % of the total space in the file system.</p> <ul style="list-style-type: none"> • <, <= <number> <p>The free space is smaller than (or the same as) <number> K M G bytes.</p>
TABLESPACE_OFFLINE (formerly TSP_OFFLINE)	Error	<p>Checks whether there are tablespaces that are offline.</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.</p>
TABLESPACE_IN_BACKUP (formerly TSP_BACKUP_MODE)	Warning	<p>This condition checks whether there are tablespaces that have the BACKUP status although BRBACKUP is not active</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.</p>
TABLESPACE_FULL (formerly TSP_FULL)	Warning	<p>This condition checks the maximum level of tablespaces on the database host. All tablespaces are checked by default against the same threshold value. However, you can define different threshold values for individual tablespaces by specifying the tablespace name in the OBJECT field of the DBCHECKORA table (transaction DB17).</p> <p>Possible test operands, threshold values, value units:</p> <ul style="list-style-type: none"> • >, >= <number> <p>The space used is larger than (or the same as) <number> % of the total space in the tablespace.</p> <ul style="list-style-type: none"> • >, >= <number> <p>The space used is larger than (or the same as) <test> K M G bytes.</p> <ul style="list-style-type: none"> • <, <= <number> <p>The free space is less than (or the same as) <number> % of the total space in the tablespace.</p> <ul style="list-style-type: none"> • <, <= <number> <p>The free space is smaller than (or the same as) <number> K M G bytes.</p>
DATA_FILE_MISSING	Warning	<p>This condition checks whether there are data files that no longer exist in the file system.</p>

25.5 BRCONNECT

(formerly FILE_MISSING)		The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
REDOLOG_FILE_MISSING (formerly REDOLOG_MISSING)	Error	This condition checks whether there are online redo log files that no longer exist in the file system. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
CONTROL_FILE_MISSING (formerly CONTROL_FILE_MISSING)	Error	This condition checks whether there are control files that no longer exist in the file system. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
DATA_FILE_MISMATCH (formerly FILE_MISMATCH)	Error	This condition checks whether there are data files that are flagged as MISSING in Oracle control file. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
INVALID_FILE_TYPE (formerly FILE_TYPE_UNKNOWN)	Error	This condition checks whether there are database files that have an illegal operating system type, for example, block raw files on Unix or compressed files on Windows. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
REDOLOG_FILE_MIRROR (formerly REDOLOG_MIRROR)	Error	This condition checks whether there are online redo log files that are not mirrored on the Oracle side. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
CONTROL_FILE_MIRROR (formerly CONTROL_MIRROR)	Error	This condition checks whether there are control files that are not mirrored on the Oracle side. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
FILE_OFFLINE (formerly DF_OFFLINE)	Error	This condition checks whether there are data files or online redo log files that are OFFLINE. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
CRITICAL_FILE	Warning	This condition examines the data files with an activated auto extend feature. A check is made to see whether the file system can be brought to overflow, due to the existing parameter setting (NEXT and MAXSIZE) during the automatic filename extension.

		The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
TOO_MANY_EXTENTS (formerly MANY_EXTENTS)	Warning	<p>This condition checks whether there are tables or indices, for which the number of allocated extents exceeds the specified threshold value. All tables or indices are checked by default against the same threshold value. However, you can define different threshold values for individual tablespaces by specifying the tablespace name in the OBJECT field of the DBCHECKORA table (transaction DB17).</p> <p>Possible test operands, threshold values, value units:</p> <ul style="list-style-type: none"> • >, >= <number> <p>More than <number> % of the maximum number of extents was already allocated.</p> <ul style="list-style-type: none"> • >, >= <number> <p>More than <number> extents were already allocated</p> <ul style="list-style-type: none"> • <, <= <number> <p>Fewer than <number> % of the maximum number of extents can still be allocated.</p> <ul style="list-style-type: none"> • <, <= <number> <p>Fewer than <number> extents can still be allocated.</p>
CRITICAL_SEGMENT (formerly CRITICAL_SEGS)	Warning	<p>This condition checks whether there are tables or indexes that can bring the tablespace to overflow when up to 5 next extents are allocated. All tables or indices are checked by default against the same threshold value. However, you can define different threshold values for individual tablespaces by specifying the tablespace name in the OBJECT field of the DBCHECKORA table (transaction DB17).</p> <p>Possible test operands, threshold values, value units:</p> <p><= <number></p> <p>The tablespace overflows when you allocate up to 1, 2, 3, 4, or 5 next extents.</p>
IN_WRONG_TABLESPACE (formerly TABLES_NOT_IN_TABLE_T ABLESPEACE)	Error	<p>This condition checks whether there are tables that not in a table tablespace or indices, which are not in an index tablespace.</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.</p>
MISSING_INDEX (formerly MISSING_INDEXES)	Error	<p>This condition checks whether there are tables that do not have any indices and are not specified in the DBDIFF table exception.</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands,</p>

25.5 BRCONNECT

		threshold values, or value units.
MISSING_STATISTICS (formerly NO_OPT_STATS)	Error	This condition checks whether there are tables or indices that do not have any statistics, although they should have these. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
HARMFUL_STATISTICS	Error	This condition checks whether there are tables or indices that have statistics, although they should not have these (for example, pool and cluster tables). The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.
PCT_INCREASE_NOT_ZERO	Error	This condition checks whether there are tables or indexes for which the PCTINCREASE storage parameter is not equal to zero. This can lead to storage fragmentation and is not suitable for the SAP System. The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.



Since the check conditions for database administration are hard-coded in BRCONNECT (known as built-in check conditions), no new test conditions can be added in the DBCHECKORA table. You can exclude individual tables and indexes or even complete tablespaces from certain checks (which run at table or index level). To do this, specify the objects in the [check_exclude](#) [Page 451] parameter:

```
check_exclude = [<owner>.<table> | [<owner>.<index> |
<tablespace> | (<object_list>)
```

You can restrict the following check conditions in this way:


TOO_MANY_EXTENTS, CRITICAL_SEGMENT, IN_WRONG_TABLESPACE,
MISSING_INDEX, MISSING_STATISTICS, HARMFUL_STATISTICS




BRCONNECT Default Conditions for Database Operations

The check conditions for database operations are specified in the control table DBCHECKORA. For more information, see [Database System Check with BRCONNECT \[Page 409\]](#).

Condition	Severity Level	Description
LAST_ARCHIVE_FAILED	Warning	This condition checks whether the last backup of the offline redo log files with BRARCHIVE failed. BRCONNECT takes the following function IDs into account: sve, cpy, ssv, svd, cpd, ssd, cps, cds The OBJECT field is not specified for this

		condition. This condition does not have test operands, threshold values, or value units.
LAST_BACKUP_FAILED	Warning	<p>This condition checks whether the last complete backup of the database with BRBACKUP failed. BRCONNECT takes the following function IDs into account:</p> <p>afd, afp, afr, afs, aft, and, anp, anr, ans, ant, ffd, ffp, ffr, ffs, fft, fnd, fnp, fnr, fns, fnt</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.</p>
LAST_STATS_FAILED	Warning	<p>This condition checks whether the last update of the optimizer statistics with BRCONNECT failed. BRCONNECT takes the following function IDs into account:</p> <p>sta, aly</p> <p>These refer to the entire database.</p> <p>The OBJECT field is not specified for this condition. This condition does not have test operands, threshold values, or value units.</p>
LAST_OPERATION_FAILED	Warning	<p>This condition checks whether the last DBA operation failed, which is identified by a function ID specified in the OBJECT field of the DBCHECKORA table.</p> <p>This condition does not have test operands, threshold values, or value units.</p>  <p>In SAP Releases 4.0 and 4.5, where the OBJECT field is not yet defined in DBCHECKORA, you can define this condition by specifying the function ID in the PARAM field. You can also use this convention in later (including current) SAP releases.</p>
ARCHIVE_TOO_OLD	Warning	<p>This condition checks whether the last successful backup of the offline redo log files with BRARCHIVE is too old. BRCONNECT takes into account the following function ids:</p> <p>sve, cpy, ssv, svd, cpd, ssd, cps, cds</p> <p>The OBJECT field is not specified for this condition. Possible test operands,</p>

25.5 BRCONNECT

		<p>threshold values, value units:</p> <p>>, >= <number> D</p> <p>The last successful backup of the offline redo log files is older than <number> day(s).</p>
BACKUP_TOO_OLD	Warning	<p>This condition checks whether the last successful complete backup of the database with BRBACKUP is too old. BRCONNECT takes into account the following function IDs:</p> <p>afd, afp, afr, afs, aft, and, anp, anr, ans, ant, ffd, ffp, ffr, ffs, fft, fnd, fnp, fnr, fns, fnt</p> <p>The OBJECT field is not specified for this condition. Possible test operands, threshold values, value units:</p> <p>>, >= <Number> D</p> <p>The last successful complete backup of the database is older than <number> day(s).</p>
STATS_TOO_OLD	Warning	<p>This condition checks whether the last successful update of the</p> <p>Optimizer statistics with BRCONNECT is too old. BRCONNECT takes the following function ID into account:</p> <p>sta, aly</p> <p>The OBJECT field is not specified for this condition. Possible test operands, threshold values, value units:</p> <p>>, >= <number> D</p> <p>The last successful update of the optimizer statistics is older than <number> day(s).</p>
OPERATION_TOO_OLD	Warning	<p>This condition checks whether the last successful DBA operation, identified by a function ID specified in the OBJECT field of the DBCHECKORA table, is too old.</p> <p>Possible test operands, threshold values, value units:</p> <p>>, >= <number> D</p> <p>The last successful DBA operation is older than <number> day(s).</p> <p></p> <p>In SAP Releases 4.0 and 4.5, where the OBJECT field is not yet defined in DBCHECKORA, you can define this condition by specifying the function ID in</p>

		the <code>PARAM</code> field. You can also use this convention in later (including current) SAP releases.
--	--	---



Since the test conditions for database operations are programmed in a specific way in BRCONNECT (known as built-in test conditions), no new check conditions can be added to the `DBCHECKORA` table. However, this is generally not necessary because operations can be monitored by the `LAST_OPERATION_FAILED` and `OPERATION_TOO_OLD` check conditions, or by specifying function IDs in the `PARAM` field.

25.5.2 Adapt Next Extents with BRCONNECT

Use

You can use [BRCONNECT \[Page 407\]](#) to adapt the next extents size. The aim is to avoid the structure of tablespaces deteriorating – that is, breaking up into a large number of small extents – because this reduces database performance.



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -next <tablespace\(s\)> \[Page 351\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Prerequisites

The database tables and indexes are assigned to one of a number of categories. The standard size of the next extent for each category is defined in the tables `TGORA` (for tables) and `IGORA` (for indexes).

Activities

When a table requires adapting, the size of the next extent is determined using an algorithm. This also makes sure that the value of `MAXEXTENTS` for a table or index is not less than the value defined in the `TGORA` or `IGORA` table.

You can exclude specified tables or indexes from this function by using the `next_exclude` parameter.

You can specify individual values for `NEXTTEXTENTS` or `MAXEXTENTS` using the `next_special` parameter.

For more information on the command line options for adapting the next extents, see [-f next \[Page 435\]](#).

See also:

[Internal Rules for Determining Next Extent Size \[Page 419\]](#)



Internal Rules for Determining Next Extent Size

This algorithm is used by BRCONNECT to determine the next extent size when a table needs adapting. For more information, see [Adapt Next Extents \[Page 419\]](#).

1. BRCONNECT works out the value of 10% of the space currently allocated to the table or index.

25.5 BRCONNECT

2. Using this value, BRCONNECT selects the **next smaller category** from the table TGORA (for tables) or IGORA (for indexes). The tables are shown below at the end of this section.
3. BRCONNECT looks up the table in the DD09L table and selects the category given. The DD09L is a data dictionary table in which most SAP tables are entered. If the table is not there, size category 0 is assumed.
4. BRCONNECT chooses the **larger** of the values from steps 2 and 3, if necessary reducing it to the value of `next_max_size`.
5. If the required space is greater than the remaining free space in the tablespace and no file in the affected tablespace permits an autoextend, BRCONNECT reduces the value from step 4 to the size of the **largest free space segment** in the tablespace.
6. BRCONNECT compares the value from step 5 with the current value of NEXTEXTENT and chooses the **larger** value.
7. If the `next_special` parameter is defined for the table, BRCONNECT always uses this instead of the value determined so far.
8. BRCONNECT reduces the value from step 7 to the next smaller multiple of 5 times the database block size, to reduce free space wastage. The smallest possible value is 5 times the database block size.
9. BRCONNECT compares the value from step 8 with the current next extent size. If there is a difference, it changes the NEXTEXTENT storage parameter of the table or index to the newly determined next extent size.

NEXT Values in TGORA/IGORA

Size category	NEXT value for table (KB)	NEXT value for indexes (KB)
0	40	40
1	160	80
2	640	160
3	2560	640
4	10240	2560
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

25.5.3 Update Statistics with BRCONNECT

Use

You can use this [BRCONNECT \[Page 407\]](#) function to update the statistics on the Oracle database for the cost-based optimizer.

By running update statistics regularly, you make sure that the database statistics are up-to-date, so improving database performance. The Oracle cost-based optimizer (CBO) uses the statistics to optimize access paths when retrieving data for queries. If the statistics are out-of-date, the CBO might generate inappropriate access paths (such as using the wrong index), resulting in poor performance.

From Release 4.0, the CBO is a standard part of the SAP System. If statistics are available for a table, the database system uses the cost-based optimizer. Otherwise, it uses the rule-based optimizer.

BRCONNECT supports update statistics for the following:

- Partitioned tables, except where partitioned tables are explicitly excluded by setting the active flag in the DBSTATC table to I. For more information, see SAP Note 424243.
- [InfoCube tables \[Page 425\]](#) for the SAP Business Information Warehouse (SAP BW)



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA functionality:

- [sapdba -statistics \[Page 353\]](#)
- [sapdba -analyze \[Page 345\]](#)
- [sapdba -delete \[Page 350\]](#)
- [sapdba -checkopt \[Page 348\]](#)

The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Integration

You can update statistics using one of the following methods:

- DBA Planning Calendar in the Computing Center Management System (CCMS)

For more information, see [Update Statistics for the Cost-Based Optimizer in CCMS \(Oracle\) \[Ext.\]](#). The DBA Planning Calendar uses the BRCONNECT commands.



We recommend you to use this approach because you can easily schedule update statistics to run **automatically** at specified intervals (for example, weekly).

- BRCONNECT, as described here
- [SAPDBA \[Page 222\]](#) – we strongly recommend you **not** to use this approach

Prerequisite

To use the CBO, make sure that the parameter `OPTIMIZER_MODE` in the Oracle initialization profile [init<DBSID>.ora \[Page 47\]](#) is set to `CHOOSE`.

Features

BRCONNECT performs update statistics using a two-step approach.

BRCONNECT:

1. Checks each table to see if the statistics are out-of-date
2. If required, updates the statistics on the table immediately after the check

25.5 BRCONNECT

For more information about how update statistics works, see [Internal Rules for Update Statistics \[Page 424\]](#).

Activities

You can influence how update statistics works by using the `-force` options. For more information, see [-f stats \[Page 436\]](#).



Unless you have special requirements, we recommend you to perform the standard update statistics, using one of the following tools to schedule it on a regular basis (for example, weekly):

- DBA Planning Calendar, as described above in "Integration."
- A tool such as cron (UNIX) or at (Windows NT) to execute the following standard call:

```
brconnect -u / -c -f stats -t all
```

This is also adequate after an upgrade of the database or SAP System. It runs using the OPS\$ user without operator intervention.

The following are also standard commands that you can use to update statistics:

- Update statistics only for tables and indexes with missing statistics
`brconnect -u / -c -f stats -t missing`
- Check and update statistics for all tables defined in the DBSTATC table
`brconnect -u / -c -f stats -t dbstatc_tab`

Example

For examples of how you can override the [internal rules for update statistics \[Page 424\]](#), see [-force with Update Statistics \[Page 422\]](#).

**-force with Update Statistics**

This section gives examples of how you can use the `-force` options to override the [internal rules for update statistics \[Page 424\]](#). For more information about the `-force` options, see [-f stats \[Page 436\]](#).



Only use these options in exceptional circumstances.

- Check and update statistics for all tables in tablespace PSAPBTAD, including pool and cluster tables
`brconnect -u / -c -f stats -t psapbtad -f allsel`
- Update statistics without check for all tables relevant to the application monitor
`brconnect -u / -c -f stats -t dbstatc_mon -f collect`
- Update statistics for a pool table specified in the DBSTATC table, to determine space usage, using method "estimate 10% rows"
`brconnect -u / -c -f stats -t atab -m EI -s P10 -f allsel,method,sample`

BRCONNECT stores the space usage data in tables DBSTATTORA (for tables) and DBSTATIORA (for indexes). Finally, the statistics are immediately deleted.

If you want to keep the statistics – for example, for test purposes – you must include the `keep` option in the command, as follows:

```
brconnect -u / -c -f stats -t rfbldg -m EI -s P10 -f
allsel,method,sample,keep
```

- Check and update statistics according to the two-phase concept:
 - a. First phase: Check statistics


```
brconnect -u / -c -f stats -t all -f nocoll
```
 - b. Second phase: Update statistics on tables identified in the first phase


```
brconnect -u / -c -f stats -t all -f nocheck
```



Deletion of Damaging Statistics

Use

This section describes how BRCONNECT deletes damaging statistics for the cost-based optimizer of the Oracle database.

Pool and cluster tables and tables that have the `ACTIVE` flag set to `N` or `R` in the `DBSTATC` control table should not normally have statistics, since such statistics can negatively affect database performance.

Activities

In the standard update statistics run, using `brconnect -f stats -t all`, BRCONNECT checks whether such damaging statistics exist and deletes them if so.

You can delete such damaging statistics immediately:

```
brconnect -u / -c -f stats -d
```

To delete statistics for other tables as well (only for test purposes), you can use the option `-f allsel`:

```
brconnect -u / -c -f stats -t sdbah,sdbad -d -f allsel
```



Verification of Table and Index Structure

Use

This section describes how you can use BRCONNECT for the Oracle database to check the internal structure of table and index blocks. This is an alternative to `DBVERIFY`. However, the scope of the checks is not the same.

Example

To check the structure of all tables and their indexes in the tablespace `PSAPBTABD`, enter the following command:

```
brconnect -u / -c -f stats -t psapbtabd -v cascade
```



This type of verification locks the tables and indexes being checked in Oracle 8.1. Therefore, only use it when database activity is low. Starting with Oracle 9, tables and indexes are no longer locked.

25.5 BRCONNECT



Internal Rules for Update Statistics

This algorithm is used by BRCONNECT to update statistics. For more information, see [Update Statistics with BRCONNECT \[Page 420\]](#).

- BRCONNECT determines the working set of tables and indexes to be checked and updated. To do this, it uses:
 - Options `-t` | `-table` and `-e` | `-exclude`, as described in [-f stats \[Page 436\]](#) (these options take priority)
 - [stats table \[Page 475\]](#) and [stats exclude \[Page 472\]](#) parameters
- If the working set contains pool, cluster or other tables that have the `ACTIVE` flag in the `DBSTATC` table set to `N` or `R`, BRCONNECT immediately deletes the statistics for these tables, because they negatively affect database performance.
- BRCONNECT checks statistics for the remaining tables in the working set, including tables that have the `ACTIVE` flag in the `DBSTATC` table set to `A` or `P`, as follows:
 - If the table has the `MONITORING` attribute set, BRCONNECT reads the number of inserted, deleted, and updated rows from the `DBA_TAB_MODIFICATIONS` table (this is available from Oracle 8.1 onwards).
 - Otherwise, BRCONNECT uses the standard method (see table below) to update statistics by using the unique index.



BRCONNECT uses the following standard method to check and update a table's statistics:

- Method and sample defined for the table in the `DBSTATC` table (has highest priority)
- Method and sample from the options `-m` | `-method` or `-s` | `-sample` of [-f stats -method \[Page 436\]](#) (takes priority) or the [stats method \[Page 473\]](#) and [stats sample size \[Page 474\]](#) parameters
- Default method and sample (has lowest priority)

The following table describes the default method:

Number of rows in table			Analysis method	Sample size
	Rows	< 10,000	C	
10,000 <=	Rows	< 100,000	E	P30
100,000 <=	Rows	< 1,000,000	E	P10
1,000,000 <=	Rows	< 10,000,000	E	P3
10,000,000 <=	Rows		E	P1

Analysis method C means compute the statistics exactly. Analysis method E means estimate the statistics using the sample size specified.

For example, "E P10" means that BRCONNECT takes an estimated sample using 10% of rows.

For the CH, CX, EH, and EX methods, histograms are created.

For the CI, CX, EI and EX methods, the structure of indexes is validated in addition to collecting statistics.

- BRCONNECT uses the number of new rows for each table in the working set, as derived in the previous step, to see if either of the following is true:

- Number of new rows is greater than or equal to number of old rows * (100 + threshold) / 100
- Number of new rows is less than or equal to number of old rows * 100 / (100 + threshold)

The standard threshold is 50, but the value defined in [-f stats -change \[Page 436\]](#) or the [stats change threshold \[Page 471\]](#) parameter is used if specified.

5. BRCONNECT immediately updates statistics after checking for the following tables:
 - Tables where either of the conditions in the previous step is true
 - Tables from the DBSTATC table with either of the following values:
 - ACTIVE field U
 - ACTIVE field R or N and USE field A (relevant for the application monitor)
6. BRCONNECT writes the results of update statistics to the DBSTATTORA table and also, for tables with the DBSTATC history flag or usage type A, to the DBSTATHORA table.
7. For tables with update statistics using methods EI, EX, CI, or CX, BRCONNECT validates the structure of all associated indexes and writes the results to the DBSTATIORA table and also, for tables with the DBSTATC history flag or usage type A, to the DBSTAIHORA table.
8. BRCONNECT immediately deletes the statistics that it created in this procedure for tables with the ACTIVE flag set to N or R in the DBSTATC table.



Update Statistics for InfoCube Tables

Use

The InfoCube tables used in SAP Business Information Warehouse (SAP BW) and Advanced Planner and Optimizer (APO) need to be processed in a special way when the statistics are being updated. Usually, statistics should be created using histograms, as described in SAP Note 129252.

Statistics for the InfoCube tables can be updated, together with other tables in a run. In this case, the statistics for the InfoCube tables are always created with histograms. You specify which tables are to be handled as InfoCube tables using the `init<DBSID>.sap` parameter [stats info_cubes \[Page 473\]](#).

Prerequisites

The control table RSNSPACE for Business Information Warehouse (BW) and Advanced Planner and Optimizer (APO) dynamically determines which tables are to be handled as InfoCube tables. The content of the control table might change in future.

Features

Tables whose names start with the following prefixes are dealt with by BRCONNECT as InfoCube tables:

```
/BIC/F*, /BIC/A9F*,
/BI0/F*, /BI0/A9F*,
/BIC/E*, /BIC/A9E*,
/BI0/E*, /BI0/A9E*,
/BIC/D*, /BIC/A9D*,
```

25.5 BRCONNECT

```

/BIO/D*, /BIO/A9D*,
/BIC/S*, /BIC/A9S*,
/BIO/S*, /BIO/A9S*,
/BIC/X*, /BIC/A9X*,
/BIO/X*, /BIO/A9X*,
/BIC/Y*, /BIC/A9Y*,
/BIO/Y*, /BIO/A9Y*,
/BIC/I*, /BIC/A9I*,
/BIO/I*, /BIO/A9I*,
/BIC/P*, /BIC/A9P*,
/BIO/P*, /BIO/A9P*,
/BIC/Q*, /BIC/A9Q*,
/BIO/Q*, /BIO/A9Q*

```

The above list provides the default value of the `init<DBSID>.sap` parameter `stats_info_cubes`, that you can use to include the following kinds of tables in the list of InfoCube tables:

- Groups with names starting with a certain prefix
- Individual tables in the list of InfoCube tables



To include the above default list, specify the keyword `DEFAULT` in the first position, as in the following example:

```
stats_info_cubes = (DEFAULT, XYZ*)
```

To suppress special handling of the InfoCube tables completely, use the keyword `NULL`:

```
stats_info_cubes = NULL
```

If certain prefixes are omitted in the parameter definition, the corresponding tables are **not** to be handled as InfoCube tables. However, we do **not** recommend you to do this.

For additional, special handling of InfoCube tables, you can use the keyword `INFO_CUBES` for the following:

- BRCONNECT with the [-f stats function \[Page 436\]](#) using the `-t | -table` and `-e | -exclude` options
- `init<DBSID>.sap` parameters:
 - [stats_table \[Page 475\]](#)
 - [stats_exclude \[Page 472\]](#)
 - [stats_dbms_stats \[Page 472\]](#)

The function of this keyword is to ensure that only InfoCube tables are processed in accordance with the selected parameter settings.

Examples

- `brconnect -u / -c -f stats -t info_cubes`

Statistics are only checked for InfoCube tables and updated, if required

- `brconnect -u / -c -f stats -t all -e info_cubes`

Statistics are checked for all tables besides InfoCube tables and updated, if necessary.

- `stats_dbms_stats = INFO_CUBES:R:4`
`brconnect -u / -c -f stats -t all`

Statistics are checked for all tables and updated, if necessary. New statistics for InfoCube tables are created with the DBMS_STATS package using row sampling and an internal parallel degree of 4.

For more information, see [stats_dbms_stats \[Page 472\]](#).

- `brconnect -u / -c -f all`

This is the default. Statistics are checked for all tables and updated, if necessary. If InfoCube tables are present and selected following the update check, statistics are generated for them using histograms.



In the current SAP standard system, all InfoCube tables are entered in control table DBSTATC with active flag "I" (ignore). This means that InfoCube tables from BRCONNECT are ignored and not processed at all. Before you can process them, the corresponding DBSTATC entries must be changed or deleted.

Alternatively, you can use option `-f allsel` to force processing of InfoCube tables, for example:

```
brconnect -u / -c -f stats -t info_cubes -f allsel
```

25.5.4 Clean Up Old Logs and Traces with BRCONNECT

Use

You can use [BRCONNECT \[Page 407\]](#) to clean up old log files, disk backups, export files, trace files and database logs. The aim is to avoid unnecessary use of disk space by deleting files that are no longer required.



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -cleanup \[Page 350\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Features

BRCONNECT cleans up the following files:

- Detailed BRARCHIVE logs in the `saparch` directory
- Detailed BRBACKUP logs in the `sapbackup` directory
- Detailed BRCONNECT logs in the `sapcheck` directory
- Detailed BRRESTORE logs in the `sapbackup` directory
- Detailed SAPDBA logs in the `sapreorg` directory
- BRBACKUP disk backups of the database files
- BRARCHIVE disk backups of the offline redo log files
- SAPDBA export dump and export script directories

25.5 BRCONNECT

- Oracle trace and audit files
- Log records in the SDBAH and SDBAD tables
- Log records in the XDB tables
- Database check results in DBMSGORA table

Activities

Using the cleanup parameters in the [Initialization Profile init<DBSID>.sap \[Page 445\]](#) you can determine how old the objects are before they are deleted. For example, see [cleanup_brarchive_log \[Page 452\]](#).

For more information on the command line options for cleaning up the log files, see [-f cleanup \[Page 432\]](#).

25.5.5 Additional BRCONNECT Functions

Use

These additional [BRCONNECT \[Page 407\]](#) functions only write messages to the standard output device.



We **strongly recommend** you to use BRCONNECT functions rather than the equivalent [SAPDBA ones \[Page 344\]](#) (if one is available, this is indicated below). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Features

- Change Passwords of Database Users
You can use this function to specify a new password for all SAP database users.
At the same time, the encrypted password in the SAPUSER table is changed. The work processes of the application server use this password to connect to the database.
For more information, see [-f chpass \[Page 432\]](#).
Use this function rather than [sapdba -alter_user \[Page 345\]](#).
- Setting Up SAP DBA Synonyms
You can use this function to specify the SAP System for database administration activities in a multi-schema database. You can then start and monitor database actions in the specified SAP System.
For more information, see [-f crsyn \[Page 434\]](#).
- Starting the Database Instance
You can start the database instance using `-f dbstart`. There are no function options for this.
Use this function rather than [sapdba -startup\(dba\) \[Page 353\]](#).
- Stopping the Database Instance
You can use this function to stop the database instance. BRCONNECT first checks whether an SAP System is still running.
For more information, see [-f dbshut \[Page 434\]](#).

Use this function rather than [sapdba -shutdown\(_abort\)](#) [Page 353].

- Determine Database State

You can use this function to determine the database state. The return codes have the following meanings:

- 0 – Database is running
- 1 – Database is stopped
- 2 – Database is in nomount or mount state
- 3 – Error, database status cannot be determined

There are no function options for this.

25.5.6 Command Options for BRCONNECT

This section describes the command options for the BRCONNECT tool.

If you use BRCONNECT with command options, these override the corresponding values in the [initialization profile init<DBSID>.sap](#) [Page 445]. To use the options, you can specify either letter indicated or the complete word.



We **strongly recommend** you to use BRCONNECT functions rather than the equivalent [SAPDBA ones](#) [Page 344]. The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Syntax

The syntax of a BRCONNECT command is:

```
brconnect [<command_options>] -f|-function <function>
[<function_options>]
```



```
brconnect -output detail -function check -default
```

Options

brconnect

```
[ -c | -confirm ]
[ -h | -help [<function>] | [version] ]
[ -l | -language E | D ]
[ -o | -output detail | process | summary | [,]time ]
[ -p | -profile <profile> ]
[ -q | -query [check] ]
[ -s | -sapsid <sid> | <sid_list> ]
[ -u | -user [<user> [/<password>]] ]
[ -V | -VERSION ]
-f|-function <function> [<function_options>]
```

See also:

25.5 BRCONNECT

[-c|-confirm \[Page 430\]](#)[-h|-help \[Page 441\]](#)[-l|-language \[Page 441\]](#)[-o|-output \[Page 441\]](#)[-p|-profile \[Page 441\]](#)[-q|-query \[Page 442\]](#)[-s|-sapsid \[Page 442\]](#)[-u|-user \[Page 442\]](#)[-V|-VERSION \[Page 443\]](#)[-f|-function \[Page 430\]](#)**-c|-confirm**

This BRCONNECT command option activates processing in unattended mode.

Input syntax: `-c | -confirm`

Default value: Confirmation required for processing to be started.

Use this option if BRCONNECT is started by an automatic scheduler such as `cron` (UNIX) or `at` (Windows NT).

**-f|-function**

This BRCONNECT command option specifies the function to be performed. You must always enter a function.



We **strongly recommend** you to use BRCONNECT functions rather than the equivalent [SAPDBA ones \[Page 344\]](#) (if one is available, this is indicated at the links below). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Input syntax: `-f`

`chpass | crsyn | dbshut | dbstart | dbstate | check | cleanup | next | stats`

Default value: None, since you must always specify a function option

Function Options

- **check**
[Checks the database system \[Page 431\]](#)
- **chpass**
[Changes the password of SAP database users \[Page 432\]](#)
- **cleanup**
[Cleans up database logs \[Page 432\]](#)
- **crsyn**

[Creates public synonyms for tables \[Page 434\]](#) used by BRARCHIVE, BRBACKUP, BRCONNECT, and SAPDBA

- **dbshut**
[Shuts down the database \[Page 434\]](#)
- **dbstart**
Starts up the database (no additional options)
- **dbstate**
Checks the database state (no additional options)
- **next**
[Adapts next extents \[Page 435\]](#)
- **stats**
[Updates optimizer statistics \[Page 436\]](#)



-f check

This BRCONNECT [function \[Page 430\]](#) checks the database system. For more information, see [Database System Check \[Page 409\]](#).



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -check \[Page 347\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Function options:

- **-d | -default**: Uses internal BRCONNECT default settings to check the database system
Input syntax: **-d**
Default value: Uses settings from the control table `DBCHECKORA`.
- **-e | -exclude**: Defines tables and indexes to be excluded from the check
Input syntax: **-e**
`[<owner>.]<table>, [<owner>.]<index>, <tablespace> | <object_list> | no_n_sap | all_part | null`
no_n_sap means that non-SAP objects (for example, Oracle dictionary objects) are excluded from the check.
all_part means that SAP partitions (such as in Business Information Warehouse and Advanced Planner and Optimizer) are excluded from the check.
null invalidates the exclusion list defined by the [check_exclude \[Page 451\]](#) parameter. This means that no tables are excluded from processing.
Default value: no exclusion
You can use this option to exclude tables or indexes with exceptional space parameters or statistics handling.
This option overrides the [check_exclude \[Page 451\]](#) parameter.
- **-o | -owner**: Defines the database owner of tables and indexes to be checked

25.5 BRCONNECT

Input syntax: `-o <owner1>[,<owner2>,...]`

Default value: All SAP owners in a multi-schema database or `SAPR3/SAP<SID>` in a standard SAP database

This option overrides the [check_owner \[Page 452\]](#) parameter.

**-f chpass**

This BRCONNECT [function \[Page 430\]](#) changes the database user password.



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -alter user \[Page 345\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Function options:

- `-o` | `-owner`: Defines for which database owner (that is, user) the password is to be changed.

Input syntax: `-o <owner1>[,<owner2>,...]`

Default value: `SAPR3/SAP<SID>` in a standard SAP database or all SAP owners in a multi-schema database

- `-p` | `-password`: Defines the password for the SAP database owner (that is, user)

Input syntax: `-p <password>`

Default value: interactive entry of the password

**-f cleanup**

This BRCONNECT [function \[Page 430\]](#) cleans up the database logs. For more information, see [Clean Up Old Logs and Traces \[Page 427\]](#).



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -cleanup \[Page 350\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Function options:

- `-a` | `-archive`: Defines the retention period in days for BRARCHIVE detail log files

Input syntax: `-a <days>`

Default: 30

This option controls which BRARCHIVE log files are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_brarchive_log \[Page 452\]](#) parameter.

- `-b` | `-backup`: Defines the retention period in days for BRBACKUP detail log files

Input syntax: `-b <days>`

Default: 30

This option controls which BRBACKUP log files are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_brbackup_log \[Page 452\]](#) parameter.

- **-c | -connect**: Defines the retention period in days for BRCONNECT detail log files
 Input syntax: **-c <days>**
 Default: 30
 This option controls which BRCONNECT log files are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_brconnect_log \[Page 452\]](#) parameter
- **-d | -dblog**: Defines the retention period in days for records in the tables SDBAH and SDBAD
 Input syntax: **-d <days>**
 Default: 100
 This option controls which records in the tables SDBAH and SDBAD are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_db_log \[Page 453\]](#) parameter.
- **-e | -expdump**: Defines the retention period in days for SAPDBA export dumps and scripts
 Input syntax: **-e <days>**
 Default: 30
 This option controls which directories and their contents for SAPDBA export dumps and scripts are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_exp_dump \[Page 453\]](#) parameter.
- **-i | -diskarch**: Defines the retention period for offline redo log files saved on disk.
 Input syntax: **-i <days>**
 Default: 30
 This option controls which offline redo log files backed up on disk are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_disk_archive \[Page 453\]](#) parameter.
- **-k | -diskback**: Defines the retention period in days for database files saved on disk.
 Input syntax: **-k <days>**
 Default: 30
 This option controls which database files backed up on disk are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_disk_backup \[Page 453\]](#) parameter.
- **-l | -limit**: Defines that **only** objects explicitly specified by other function options are to be cleaned up
 Input syntax: **-l**
 Default: Clean up all database logs
- **-m | -msg**: This parameter defines the retention period in days for the alert messages from the database check runs. The messages are deleted from the DBMSGORA table when the retention period has expired.
 This option overrides the [cleanup_check_msg \[Page 452\]](#) parameter.
- **-o | -owner**: Defines the database owner of SDBAH, SDBA, DBMSGORA, and XDB tables to be processed by BRCONNECT cleanup function.
 Input syntax: **-o <owner1>[,<owner2>, ...]**

25.5 BRCONNECT

Default: `SAPR3/SAP<SID>` in a standard SAP database or all SAP owners in a multi-schema database

This option overrides the [cleanup_owner \[Page 453\]](#) parameter.

- `-r` | `-restore`: Defines the retention period in days for BRRESTORE detail log files

Input syntax: `-r <days>`

Default: 30

This option controls which BRRESTORE log files are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_brrestore_log \[Page 452\]](#) parameter.

- `-s` | `-sapdba`: Defines the retention period in days for SAPDBA log files

Input syntax: `-s <days>`

Default: 30

This option controls which SAPDBA log files are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_sapdba_log \[Page 454\]](#) parameter.

- `-t` | `-trace`: Defines the retention period in days for Oracle trace and audit files

Input syntax: `-t <days>`

Default: 30

This option controls which Oracle trace and audit files are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_ora_trace \[Page 453\]](#) parameter.

- `-x` | `-xdb`: Defines the retention period in days for records in the XDB tables

Input syntax: `-x <days>`

Default: 100

This option controls which records in the XDB tables are deleted by the BRCONNECT cleanup function. It overrides the [cleanup_xdb_log \[Page 454\]](#) parameter.

**-f crsyn**

This BRCONNECT [function \[Page 430\]](#) creates public synonyms for the SAP tools.

Function options:

`-o` | `-owner`: Defines the database owner pointed to by the public synonyms used by BRARCHIVE, BRBACKUP, BRCONNECT, and SAPDBA.

Input syntax: `-o <owner>`

Default value: `SAPR3/SAP<SID>`

**-f dbshut**

This BRCONNECT [function \[Page 430\]](#) shuts down the database in immediate mode.



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -shutdown\(abort\) \[Page 353\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Function options:

-f | **-force**: Shuts down the database, whether or not the SAP user is connected.

Input syntax: **-f**

Default value: If the SAP user is connected to the database, then it is not shut down.



-f next

This BRCONNECT [function \[Page 430\]](#) adapts the next extents of database tables. For more information, see:

- [Adapt Next Extents \[Page 419\]](#)
- [Algorithm for Determining Next Extent Size \[Page 419\]](#)



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA one, [sapdba -next <tablespace\(s\)> \[Page 351\]](#). The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Function options:

- **-e** | **-exclude**: Defines tables and indexes to be excluded from adapting next extents

Input syntax: **-e**

[<owner>.]<table>, [<owner>.]<index>, <tablespace> | <object_list>
| **all_part** | **null**

Default: No exclusion, process all selected objects

all_part means that SAP partitions (such as in Business Information Warehouse and Advanced Planner and Optimizer) are excluded from the check.

null invalidates the exclusion list defined by the [next_exclude \[Page 461\]](#) parameter. This means that no tables or indexes are excluded from processing.

You can use this option to exclude from processing tables or indexes with exceptional space parameters. This option overrides the [next_exclude \[Page 461\]](#) parameter.

- **-f** | **-force**: Forces the next extent size of all selected tables and indexes to be reduced to the maximum free extent size in the tablespace or to the maximum next extent size (**next_max_size** parameter) if required.

Input syntax: **-f** **free** | **max** | **both** | **nocasc**

Default: **both**

The next extent size is not normally reduced, but only rounded down to the next multiple of 5 times the database block size.

- **free**: reduces the next extent size to the maximum free extent size in the tablespace
- **max**: reduces the next extent size to the value of parameter **next_max_size**
- **both**: reduces the next extent size to the smaller of the values **free** and **max**
- **nocasc**: With this option, you can prevent the check and, if necessary, the adjustment of NEXT extents from being implicitly performed for all indexes for the selected tables. This option is only provided for exceptional situations.

- **-l** | **-limit**: Defines the maximum number of next extents (**MAXEXTENTS**)

Input syntax: **-l** <count>

25.5 BRCONNECT

Default: Settings from tables TGORA and IGORA

0 means unlimited number of extents

This option overrides the [next_limit_count \[Page 461\]](#) parameter.

- **-m | -max**: Defines the maximum size for next extents, which must **not** be exceeded when a next extent is adapted

Input syntax: **-m <size>**

Default: 2 GB - 5 * <database block size>

0 means unlimited next extent size

You can specify this option in KB, MB, or GB. This option overrides the [next_max_size \[Page 461\]](#) parameter.

- **-o | -owner**: Defines the database owner of tables and indexes to be processed

Input syntax: **-o <owner1> [, <owner2>, ...]**

Default: **SAPR3/SAP<SID>** in a standard database or all SAP owners in a multi-schema database.

This option overrides the [next_owner \[Page 462\]](#) parameter.

- **-s | -special**: Defines special sizes of next extent and maximum number of next extents for exceptional tables and indexes

Input syntax: **-s | -special**

[<owner>.]<table>:<size>[/<limit>] | [<owner>.]<index>:<size>[/<limit>] | <special_list> | all_sel:<size>[/<limit>] | null

Default: Derived from [internal rules for determining next extent size \[Page 419\]](#)

This option overrides the [next_special \[Page 462\]](#) parameter.

- **<size>**: size of next extent for specified table or index (NEXTTEXTENT)
- **<limit>**: maximum number of next extents for specified table or index (MAXEXTENTS)
- **all_sel**: sets NEXTTEXTENT and MAXEXTENTS attributes to a certain value for **all** the database objects selected using the **-t** function option or the **next_table** parameter (see below). This option is provided for exceptional situations.
- **null** invalidates the special list defined by the [next_special \[Page 462\]](#) parameter. It means that **no** tables or indexes are to be processed in a special way.

- **-t | -table**: Defines the database objects (that is, tables, indexes, or tablespaces) to be processed

Input syntax: **-t**

all | special | [<owner>.]<table> | [<owner>.]<index> | <tablespace> | <object_list>

Default: All objects of selected owners

- **all**: All SAP tables and indexes
- **special**: Only tables and indexes defined in the [next_special \[Page 462\]](#) parameter

This option overrides the [next_table \[Page 462\]](#) parameter.



-f stats

This BRCONNECT [function \[Page 430\]](#) updates optimizer statistics. For more information, see [Update Statistics \[Page 420\]](#).



We **strongly recommend** you to use this BRCONNECT function rather than the equivalent SAPDBA ones:

- [sapdba -statistics \[Page 353\]](#)
- [sapdba -analyze \[Page 345\]](#)
- [sapdba -delete \[Page 350\]](#)
- [sapdba -checkopt \[Page 348\]](#)

The reason is that we will in future no longer develop the SAPDBA functions, whereas BRCONNECT functions will always be fully up-to-date.

Function options:

- **-b | -bucket**: Defines the number of buckets in histograms
 Input syntax: `-b <count>`
 Default: 75
 This option overrides the [stats_bucket_count \[Page 471\]](#) parameter.
- **-c | change**: Changes threshold for the percentage of inserted or deleted rows causing update statistics
 Input syntax: `-c <threshold>`
 Default: 50
 This option overrides the [stats_change_threshold \[Page 471\]](#) parameter.
- **-d | -delete**: Deletes only damaging table and index statistics
 Input syntax: `-d`
 Default: Collect outdated and delete damaging statistics
 You can use this option to delete statistics for pool and cluster tables, and for tables specified in the DBSTATC control table with the active flags set to N or R.
- **-e | -exclude**: Defines tables and indexes to be excluded from update statistics
 Input syntax: `-e`
`[<owner>.]<table>,[<owner>.]<index>,<tablespace>|<object_list>|info_cubes|null`
`info_cubes` excludes check and update statistics for InfoCube tables.
`null` invalidates the exclusion list defined by the [stats_exclude \[Page 472\]](#) parameter. This means that no tables or indexes are excluded from processing.
 Default: no exclusions
 You can use this option to exclude tables or indexes with exceptional statistics handling. This option overrides the [stats_exclude \[Page 472\]](#) parameter.
- **-f | -force**: Forces a specific action while updating statistics
 Input syntax: `-f`
`[allsel] [,collect] [,history] [,keep] [,limit] [,method] [,nocasc] [,nocheck] [,nocoll] [,precision] [,sample]`
 Default: [Internal rules \[Page 424\] determine the update statistics method \[Page 424\]](#)

25.5 BRCONNECT

- **allsel**: Updates statistics for all selected objects (option -t or the [stats_table](#) parameter), including pool and cluster tables, or deletes statistics for selected objects, including non-pool and non-cluster tables
- **collect**: Update statistics without checking them first
- **history**: Stores the results of update statistics in the history tables DBSTATHORA and DBSTAIHORA, also for tables specified in the DBSTATC control table
- **keep**: Does not delete statistics after updating them for pool and cluster tables (option -f allsel) or for tables with the active flag set to N or R in DBSTATC
- **limit**: Forces hard processing time limit defined in option -l or [stats_limit_time \[Page 473\]](#) parameter. Working threads are aborted.
- **method**: Uses the method defined in option -m or [stats_method \[Page 473\]](#) parameter, also for tables specified in DBSTATC. See [stats_method \[Page 473\]](#) parameter
- **nocasc**: Prevents update statistics from being implicitly performed for all indexes of the selected tables. Only use this option in exceptional situations.
- **nocheck**: Does not check statistics. Instead, determine which tables to update statistics for by using the check results from the **previous** BRCONNECT run with -f nocoll. This is part of the two-phase concept for update statistics.
- **nocoll**: Checks statistics only by analyzing the primary index. Statistics are updated for tables that need new statistics in the **next** BRCONNECT run with -f nocheck. This is part of the two-phase concept for update statistics.
- **precision**: Forces minimum precision (that is, sample size) defined in option -s or [stats_sample_size \[Page 474\]](#) parameter for all tables if statistics are collected with method E.
- **sample**: Uses sample size defined in option -s or [stats_sample_size \[Page 474\]](#) parameter, also for tables specified in the control table DBSTATC
- **-h | -history**: Stores the results of updating statistics in the history tables DBSTATHORA and DBSTAIHORA for tables not specified in DBSTATC.

Input syntax: -h

Default: no history records are saved

The history data in the tables DBSTATHORA and DBSTAIHORA is used by the application monitor.

- **-l | -limit**: Defines the processing time limit in minutes for updating statistics

Input syntax: -l <minutes>

Default: 0, no limit

You can use this parameter to terminate long-running update statistics jobs after a certain period of time. The processing terminates after statistics have been collected for the current table or index (this is the "soft limit"). If you set the option -f limit (see above), processing terminates immediately (this is the "hard limit").

This option overrides the [stats_limit_time \[Page 473\]](#) parameter.

- **-m | -method**: Defines the method for updating statistics for tables that are not specified in the control table DBDSTATC.

Input syntax: -m E | EH | EI | EX | C | CH | CX | E= | C= | =H | =I | =X | +H | +I

Default: [Internal rules \[Page 424\]](#) determine the update statistics method

This option overrides the [stats_method \[Page 473\]](#) parameter.

E: Estimates

EH: Estimates with histograms

EI: Estimates with index validation

EX: Estimates with histograms and index validation

c: Computes

CH: Computes with histograms

CI: Computes with index validation

CX: Computes with histograms and index validation

E=: Forces estimate for all tables, including tables in DBSTATC control table. Option `-f method` must be set

C=: Forces compute for all tables, including tables in DBSTATC control table. Option `-f method` must be set

=H: Forces collect statistics with histograms for all tables, including tables in DBSTATC control table. Option `-f method` must be set

=I: Forces collect statistics with index validation for all tables, including tables in DBSTATC control table. Option `-f method` must be set

=X: Forces collect statistics with histograms and index validation for all tables, including tables in DBSTATC control table. Option `-f method` must be set

+H: Forces collect statistics with histograms for all tables, including tables in DBSTATC control table in addition to index validation, if specified in DBSTATC control table. Option `-f method` must be set.

+I: Forces collect statistics with index validation for all tables, including tables in DBSTATC control table in addition to index validation, if specified in DBSTATC control table. Option `-f method` must be set.

- **-o | -owner**: Defines the database owner of tables and indexes for updating statistics

Input syntax: `-o <owner1>[,<owner2>, ...]`

Default: `SAPR3/SAP<SID>` in a standard SAP database or all SAP owners in a multi-schema database

This option overrides the [stats_owner \[Page 474\]](#) parameter.

- **-p | -parallel**: Defines the number of parallel threads for updating statistics

Input syntax: `-p <number>`

Default: 1

For example, you can set this parameter to the number of CPUs to speed up update statistics.

This option overrides the [stats_parallel_degree \[Page 474\]](#) parameter.

- **-r | -retain**: Skips check and update of statistics of tables for which statistics were checked or updated in the specified time period

Input syntax: `-r <days>|last`

25.5 BRCONNECT

Default: `last`

- `<days>`: Skips tables and indexes for which statistics were checked or updated in the last `<days>` days
- `last`: Skips tables and indexes for which statistics were checked or updated in the last BRCONNECT run. You can use this option to restart an aborted BRCONNECT run of update statistics.
- `-s` | `-sample`: Defines the sample size in percentage or thousands of rows for updating statistics with method E for tables that are not specified in the `DBSTATC` control table.

Input syntax: `-s P<p> | R<r>`

Default: [Internal rules \[Page 424\]](#) determine the update statistics method

This option overrides the [stats_sample_size \[Page 474\]](#) parameter.

- `P<p>`: Percentage of rows
- `R<r>`: Number of thousand rows
- `-t` | `-table`: Defines the objects to be processed by update statistics.

Input syntax: `-t`

`all | all_ind | missing | dbstatc_tab | dbstatc_mon | dbstatc_mona`
`| [<owner>.]<table> | [<owner>.]<index> | <tablespace> | <object_list> | i`
`nfo_cubes`

This option overrides the [stats_table \[Page 475\]](#) parameter.

- `all`: All SAP tables and indexes
- `all_ind`: Processes all indexes only. For example, you can use this to create the space statistics for all indexes.
- `missing`: Only tables and indexes with missing statistics
- `dbstatc_tab`: Only tables specified in the `DBSTATC` control table
- `dbstatc_mon`: Only tables specified in the `DBSTATC` control table that are relevant for the application monitor
- `dbstatc_mona`: Only application tables specified in the `DBSTATC` control table that are relevant for the application monitor
- `info_cubes` checks statistics only for InfoCube tables and updates them if necessary.
- `-v` | `-validate`: Validates a table or index structure, but no update statistics is performed

Input syntax: `-v table | index | cascade | index_store | cascade_store`

Default: `cascade`

- `table`: Validate internal structure of table blocks
- `index`: Validate internal structure of index blocks
- `cascade`: Validate internal structure of table and index blocks, including relation between index and data rows
- `index_store`: Same as `index`, but also store the statistical values, determined during the validation of the index structures, in the `DBSTATIORA` table, so that they are available for the application monitor (ST07).

- **cascade_store**: Same as **cascade**, but also store the statistical values, determined during the validation of the index structures, in the **DBSTAIHORA** table, so that they are available for the application monitor (ST07).

This function can run for a long time and locks the table and indexes in Oracle 8. Therefore, we recommend performing it when there is little processing on the database. However, in Oracle 9 it does **not** lock tables and indexes.



-h|-help

This BRCONNECT command option provides help information.

Input syntax: **-h** [**<function>**] | [**version**]

Default value: Display help information about all BRCONNECT functions.

Using this option, you can obtain an overview of BRCONNECT functions.

- **<function>**: Displays help information about main options and specified functions only.
- **version**: Displays detailed information on the versions of the program modules.



-l|-language

This BRCONNECT command option sets the language for messages.

Input syntax: **-l** **E** | **D**

Default value: **E**

You can determine whether you want the 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.



-o|-output

This BRCONNECT command option controls the information written to the detail log.

Input syntax: **-o** [**detail** | **process** | **summary**] [, **time**]

Default value: **process**

- **detail**: Writes detailed processing information to the log file. The log file is then comprehensive, which can be helpful to investigate problems.
- **process**: Writes detailed information to the log file, including the analysis methods and sample sizes for updating the index statistics while statistics are being updated. These are used to determine whether tables need new statistics.
- **summary**: Writes only summary information and total counts to the log file. This can be useful for creating initial statistics to avoid a large log file.
- **time**: Generates additional time stamps that enable you to determine the time required for the individual operations.



-p|-profile

This BRCONNECT command option defines the profile name.

25.5 BRCONNECT

Input syntax: **-p** <profile>

Default value: **init<DBSID>.sap**

This profile is contained in directory <ORACLE_HOME>/dbs (UNIX) or <ORACLE_HOME>\database on Windows NT.

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

This BRCONNECT command option sets the query mode. No processing is started.

Input syntax: **-q** [**check**]

Default value: start processing

With this option, BRCONNECT displays information about the work to be done (for example, the number of database objects to be processed) by the selected function.

- **check**: displays objects that would be changed by the function (for example, the objects for which NEXT extents would be adapted by the **-f next** function)



-s|-sapsid

This BRCONNECT command option defines the SAP System ID for the objects to be processed.

Input syntax: **-s** <SAPSID1> [, <SAPSID2> , ...]

Default value: all SAP tables and indexes in the database

This parameter can be used for SAP objects with the owner name in the form SAP<SID>. For the standard SAP owner SAPR3 use R3 as the system ID.

This option is only relevant for a multi-schema database. In general, this option has the same meaning as the option **-o|-owner**, which we recommend you to use. **-o|-owner** is used in the BRCONNECT functions [-f check \[Page 431\]](#), [-f next \[Page 435\]](#), [-f stats \[Page 436\]](#).



-u|-user

This BRCONNECT command option defines the user name and password used by the SAP tool to log on to the database.

Input syntax: **-u** [<user> [/<password>]]

Default value: **system/manager**

If you only enter **-u** , an interactive query of the user name and the password is performed by the SAP tool. You can enter the user name and the password separately (only enter the user name or the option **-u <user>**). The tool 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:

```
brconnect -c -u -f stats <<END
<user>/<password>
END
```

However, use this command only if the option `-c` is active.



If you are working with an OPS\$ user, use the following call:

```
brconnect -u / -c -f stats
```

In this case, BRCONNECT tries to log on to the database as OPS\$ user (see Oracle documentation and information in the SAP Service Marketplace). The OPS\$ user must be defined in the database and have at least SYSOPER authorization and SAPDBA role. With this method, it is not necessary to specify the password when calling BRCONNECT.



-V|-VERSION

This BRCONNECT command option displays detailed information on the program version and patches.

Input syntax: `-v`

25.5.7 BRCONNECT Logs

For more information, see:

- [Names of the BRCONNECT Detail Logs \[Page 443\]](#)
- [BRCONNECT Detail Log \[Page 444\]](#)
- [BRCONNECT Summary Log \[Page 444\]](#)



Names of the BRCONNECT Detail Logs

Every BRCONNECT detail log contains a name with the following format:

`c<encoded timestamp>.<ext>`

The first characters indicate the encoded time the restore was performed (action ID). The extension (function ID) indicates the type of processing. The logs are stored in the `sapcheck` directory.

BRCONNECT only writes logs for the functions [check \[Page 431\]](#), [cleanup \[Page 432\]](#), [next \[Page 435\]](#) and [stats \[Page 436\]](#).

Possible function IDs:

- `.chk`: [Check database system \[Page 431\]](#), function `-f check [Page 431]`
- `.cln`: Clean up database log, function `-f cleanup [Page 432]`
- `.nxt`: Adapt next extents, function `-f next [Page 435]`
- `.sta`: Check and update statistics, function `-f stats [Page 436]`
- `.dst`: Delete damaging statistics, function `-f stats -d [Page 436]`
- `.opt`: Check statistics only, function `-f stats -f nocoll [Page 436]`
- `.aly`: Collect statistics for tables with outdated statistics, function `-f stats -f nocheck [Page 436]`
- `.vst`: Verify table and index structure, function `-f stats -v [Page 436]`

25.6 BRTOOLS

- `.quc`: Determine objects to be processed for a given function, command option [-q|-query](#) [Page 442]

**BRCONNECT Detail Log**

The detail log file contains information about the actions that were performed by BRCONNECT:

- The relevant parameters of initialization profile `init<DBSID>.sap` that were set during the BRRESTORE run
- Information about the numbers of objects to be processed
- For the check function, the conditions to be checked
- Processing details:
 - [Check function](#) [Page 431]: Alert conditions detected
 - [Cleanup function](#) [Page 432]: Database logs that were deleted
 - [Next function](#) [Page 435]: Tables and indexes for which the next extent was adapted
 - [Stats function](#) [Page 436]: Tables and indexes for which statistics were collected
- Summary information and total counts of objects processed

**BRCONNECT Summary Log**

You can display a brief entry for each restore in the summary log `conn<DBSID>.log`. The logs are stored in the `sapcheck` directory. The entries in the file provide the following information about each function using BRCONNECT:

- Action ID (encoded timestamp of the log name)
- Function ID (extension of the log name)
- Timestamp (date, time) specifying the start of the function
- Timestamp (date, time) specifying the end of the function
- Return code
- BRCONNECT version

25.6 BRTOOLS

Use

BRTOOLS is a tool that you cannot start yourself. Instead, it is started internally by BRBACKUP, BRARCHIVE, and BRRESTORE.

Features

- Backup verification. The saved files are compared with their originals.
See [-w|-verify](#) [Page 376]
- The option `-c force`. This denies the commands that are started (such as `cpio`) access to the console.
See [-c|-confirm](#) [Page 366]
- [Parallel backup](#) [Page 113] to a remote host (`backup_dev_file = pipe`).

25.7 Initialization Profile init<DBSID>.sap

Definition

The initialization profile `init<DBSID>.sap` contains parameters that influence how the SAP tools perform various functions. It is usually stored in directory `<ORACLE_HOME>/dbs` (UNIX) or `<ORACLE_HOME>\database` (Windows).

Use

To configure the SAP tools BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT, you must use the initialization profile `init<DBSID>.sap`. You can edit the file with a text editor. If you do not make any changes, the SAP tools use the default values for the parameters.

Before you use one of the SAP tools, 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.

Structure

The parameters and values in profile `init<DBSID>.sap` look as follows:

```
<Parameter> = <value> | (<value_list>)
```

where `value_list` = `<value_1>,<value_2>`

You separate the individual values in a value list by commas, and enclose the entire list in parentheses. You can use blanks between any symbols of such commands. If necessary, you can continue the parameter values can be continued on the next line (the line break is then treated as a blank). If a parameter value contains special characters such as space or \$, you must enclose the value in double quotes, as in the following example:

```
rewind = "mt -f $ rewind"
```

If you are unsure, 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`.

Integration

You can override many of the parameters in `init <DBSID>.sap` by using a command option when you call BRBACKUP, BRARCHIVE, BRCONNECT, or BRRESTORE. For more information, see:

- [Effects of the Command Options \[Page 359\]](#)
- [Command Options for BRBACKUP \[Page 363\]](#)
- [Command Options for BRARCHIVE \[Page 384\]](#)
- [Command Options for BRCONNECT \[Page 429\]](#)
- [Command Options for BRRESTORE \[Page 398\]](#)

Changes to parameter values do not take effect until you call the corresponding tool.

25.7.1 archive_copy_dir

This parameter defines the directory used by BRARCHIVE to back up the offline redo log files to a local disk.

25.7 Initialization Profile init<DBSID>.sap

Syntax: `archive_copy_dir = <dir>`

Default: First value of the parameter [backup_root_dir \[Page 450\]](#)

Required value: A directory in which the offline redo log files should be archived.

This is generally only required when you are working with a two-phase archiving procedure, which is similar to a [Two-Phase Backup \[Page 109\]](#). The offline redo log files are first collected in a directory on a disk and then written to tape using [BRARCHIVE option -a \[Page 385\]](#) or external means. If external tools are used, the user is responsible for archiving to tape and for restoring from the tape to the disk, if this become necessary in connection with a recovery.



BRARCHIVE archiving to disk should only be used in the situations mentioned above. In all other cases the offline redo log files must always be archived to tape. Archiving to disk is not a substitute for archiving the offline redo log files to tape.



Archiving to disk and tape cannot be combined during a single BRARCHIVE run.



When you archive the offline redo log files to disk, you may only use the BRARCHIVE options `-s`, `-sd` and `-ds`.

25.7.2 archive_function

This parameter defines the type of archiving with BRARCHIVE.

Syntax: `archive_function =`
`save|second_copy|delete_saved|deleted_copied|save_delete|second_copy_delete|double_save|double_save_delete|copy_save|copy_delete_save`

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.
- **deleted_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 parameters 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.

25.7.3 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 \[Page 445\]](#) for a backup to a local disk.

Syntax: `archive_stage_dir = <dir>`

Default: First value of the parameter [stage_root_dir \[Page 471\]](#)

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 \[Page 105\]](#).

25.7.4 backup_dev_type

Determines the backup medium that you want to use.

Syntax: `backup_dev_type = disk|tape|pipe|tape_auto|pipe_auto|tape_box|pipe_box|disk_copy|disk_standby|util_file|util_file_online|stage|stage_copy|stage_standby|rm an_prep|rman_util|rman_disk|rman_stage`

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 \[Page 464\]](#), [remote_user \[Page 464\]](#), [copy_in_cmd \[Page 456\]](#), and [copy_out_cmd \[Page 456\]](#). 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

25.7 Initialization Profile init<DBSID>.sap

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`).

For all tape device types, 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 \[Page 111\]](#).
- **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 \[Page 163\]](#).
- **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. This value can also be set for an offline backup, in which case the backup is not stopped before calling BACKINT. Instead, it is stopped when the first file is about to be backed up by BACKINT and started again after the last file has been saved.

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 [External Backup Programs \[Page 180\]](#).

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 \[Page 105\]](#).
- **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 `stage_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 save sets is determined before a RMAN backup to tape with file multiplexing.
See [RMAN Save-Set Grouping \[Page 151\]](#).
- **rman_util**: RMAN backup in combination with a backup library and the backup tool of another manufacturer. BACKINT provides an interface to the external backup tool and is also used to back up profiles, log files and the control file.
See [RMAN Backup with an External Backup Library \[Page 144\]](#).
- **rman_disk**: Backup to local disk with an external backup library and RMAN, but without BACKINT.
- **rman_stage**: Backup to remote disk with an external backup library and RMAN, but without BACKINT.

25.7.5 backup_mode

This parameter is used by BRBACKUP to determine the scope of the backup activity.

Syntax: `backup_mode =`
`all | all_data | full | incr | <tablespace> | <file_ID> | <file_ID1>-`
`<file_ID2> | <generic_path> | sap_dir | ora_dir | <object_list>`

Default: `all`

Possible values:

- **all**: Whole database backup using BRBACKUP.
- **all_data**: Save the files of all tablespaces which are not pure index tablespaces.
- **full**: Full database backup (level 0). See [Incremental Backup \[Page 83\]](#)
- **incr**: Incremental backup with RMAN. See [Incremental Backup \[Page 83\]](#).
- **<tablespace>**: The files of the specified tablespaces are backed up by BRBACKUP.
- **<file_ID>**: The file with this file ID are backed up. 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 back up all existing online redo log files.
- **<file_ID1>-<file_ID2>**: The files in this interval are backed up.
- **<generic_path>**: By entering a full path, you can save database files, non-database files, or the specified directory. By entering a generic path, you can save database data files whose name starts with that path. In this case, the path must contain at least the `SAPDATA_HOME` directory and an additional generic specification (for example, `sapdata<n>`) in the path.
- **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 tools.



You can only use this option when saving to tape and when not performing 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

25.7 Initialization Profile `init<DBSID>.sap`

`<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 tools.



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, as described in [Initialization Profile `init<DBSID>.sap` \[Page 445\]](#). This list can also include the key word **all**. However, we recommend processing database files and non-database files separately.



For UNIX systems: Start BRBACKUP to save the SAP or 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.



The `root` user must have the environment of the corresponding `ora<sid>` user for BRBACKUP to be started successfully.

25.7.6  `backup_root_dir`

This is the parameter that is used only by BRBACKUP to identify the directories in which database backups to disk are performed. 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.

Syntax: `backup_root_dir = <dir> | (<dir_list>)`

Default: `<SAPDATA_HOME>/sapbackup`

Required value: Directories to which the backups performed on disk should be written.

If you do not have enough storage space in one of the directories on your 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 \[Page 105\]](#)

25.7.7  `backup_type`

This BRBACKUP parameter identifies the type of the database backup.

Syntax: `backup_type =`
`online | online_cons | offline | offline_force | offline_standby | online_split | offline_split | offline_stop`

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 standby database configuration. See [Standby Database \[Page 162\]](#).
- **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 [Split-Mirror Disk Backup \[Page 170\]](#).
- **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 SAP System is running during the entire split mirror backup. Restriction: No transactions can be performed during the short period of time that the database is shut down. This backup mode is only relevant for the 'split mirror' configuration. For more information, see [Split-Mirror Disk Backup \[Page 170\]](#).
- **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. For more information, see [Standby Database \[Page 162\]](#).

See also:

[Backup Mode \[Page 81\]](#).

25.7.8 check_exclude

This parameter defines the database objects (that is, tables, indexes, and tablespaces) to be excluded from the [BRCONNECT check function \[Page 431\]](#).

Syntax: `check_exclude =`
`[<owner>.]<table> | [<owner>.]<index> | <tablespace> | (<object_list>) | non_sap | all_part`

Default: No exclusion, all SAP tables and indexes are checked

non_sap means that non-SAP objects are excluded from the check (for example, Oracle dictionary objects).

all_part means that SAP partitions (such as in Business Information Warehouse and Advanced Planner and Optimizer) are excluded from the check.

You can use this parameter to exclude tables or indexes with exceptional space parameters or statistics handling from the checks.

25.7 Initialization Profile init<DBSID>.sap

25.7.9 check_owner

This parameter defines the database owner of tables and indexes to be checked by the [BRCONNECT check function \[Page 431\]](#).

Syntax: `check_owner = <owner> | (<owner_list>)`

Default: `SAPR3/SAP<SID>` in a standard SAP database or all SAP owners in a multi-schema database

You can use this parameter to limit the check to objects of selected SAP owners (that is, systems).

25.7.10 cleanup_brarchive_log

This parameter defines the retention period in days for BRARCHIVE detail log files, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_brarchive_log = <days>`

Default: 30

25.7.11 cleanup_brbackup_log

This parameter defines the retention period in days for BRBACKUP detail log files, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_brbackup_log = <days>`

Default: 30

25.7.12 cleanup_brconnect_log

This parameter defines the retention period in days for BRCONNECT detail log files, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_brconnect_log`

Default: 30

25.7.13 cleanup_brrestore_log

This parameter defines the retention period in days for BRRESTORE detail log files, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_brrestore_log = <days>`

Default: 30

25.7.14 cleanup_check_msg

This parameter defines the retention period in days for the alert messages in the `DBMSGORA` table from the database check runs using [BRCONNECT check function \[Page 431\]](#). The messages are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_check_msg = <days>`

Default: 100

25.7.15 cleanup_db_log

This parameter defines the retention period in days for records in the SDBAH and SDBAD tables, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_db_log = <days>`

Default: 100

25.7.16 cleanup_disk_archive

This parameter defines the retention period in days for offline redo log files saved on disk, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_disk_archive = <days>`

Default: 30

25.7.17 cleanup_disk_backup

This parameter defines the retention period in days for database files saved on disk, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_disk_backup = <days>`

Default: 30

25.7.18 cleanup_exp_dump

This parameter defines the retention period in days for SAPDBA export dumps and scripts, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_exp_dump = <days>`

Default: 30

25.7.19 cleanup_ora_trace

This parameter defines the retention period in days for Oracle trace and audit files, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_ora_trace = <days>`

Default: 30

25.7.20 cleanup_owner

This parameter defines the database of objects (that is, tables, indexes, and tablespaces) to be excluded from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_owner = <owner> | (<owner_list>)`

25.7 Initialization Profile init<DBSID>.sap

Default: `SAPR3/SAP<SID>` in a standard SAP database or all SAP owners in a multi-schema database

You can use this parameter to limit the cleanup to objects of selected SAP owners (that is, systems).

25.7.21 cleanup_sapdba_log

This parameter defines the retention period in days for SAPDBA detail log files, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_sapdba_log = <days>`

Default: 30

25.7.22 cleanup_xdb_log

This parameter defines the retention period in days for records in the XDB tables, which are excluded for the specified period from the [BRCONNECT cleanup function \[Page 432\]](#).

Syntax: `cleanup_xdb_log = <days>`

Default: 100

25.7.23 compress

Define whether files are to be compressed (software or hardware compression).

Syntax: `compress = no|yes|hardware|only`

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 does not activate hardware compression. It is merely information for BRBACKUP/BRARCHIVE. You also have to configure your backup device accordingly. This value is also supported for backups on disk with hardware-compressing file systems on Windows NT and AIX operating systems.
- **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 455].

25.7.24 `compress_cmd`

Provides the command to be used for software file compression (if activated using the compression parameter such as: `compress = yes`).

Syntax: `compress_cmd = <cmd>`

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.

25.7.25 `compress_dir`

Directories in which file compression will be performed.

Syntax: `compress_dir = <dir> | (<dir_list>)`

Default value: The directories 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.

25.7 Initialization Profile init<DBSID>.sap



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 \[Page 459\]](#).



If you check the readability and completeness of the backup using [-w|-verify \[Page 376\]](#), the files are restored back to the directories 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 \[Page 459\]](#)).

25.7.26  `copy_in_cmd`

Defines the command used to read data from remote tape devices.

Syntax: `copy_in_cmd = <cmd>`

Default value: None

You must enter a value before you switch to “remote piping,” that is, `backup_dev_type = pipe|pipe_auto`. For more information, see [backup_dev_type \[Page 447\]](#).

This parameter specification is only useful in combination with parameters `remote_host` and `remote_user` for reading from remote systems.



```
copy_in_cmd = "dd bs=50 if=$",
```

However, you can also use higher blocking to improve performance. The \$ character is replaced by the device address. Do not forget to enclose the parameter specifications composed of several values in double quotes.

25.7.27  `copy_out_cmd`

Defines the command used to write data to remote tape devices.

Syntax: `copy_out_cmd = <cmd>`

Default: None

You must enter a value before you switch to “remote piping”.

This parameter specification is only useful in combination with the parameter `remote_host` and `remote_user` for performing backups on remote systems.



```
copy_out_cmd = "dd bs=50k of=$",
```

You can also use higher blocking to improve performance. The \$ character is replaced by the device address. Do not forget to enclose the parameter specifications composed of several values in double quotes.

25.7.28  `cpio_disk_flags`

Flags for cpio commands used by BRBACKUP if directories are copied to disk.

Syntax: `cpio_disk_flags = <flags>`

Default: `-pdcu`



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.

25.7.29 `cpio_flags`

Flags for cpio commands that the SAP tools use.

Syntax: `cpio_flags = <flags>`

Default: `-oVB`

The option `B` causes a block size of 5KB to be used. You can also use a higher block size, such as 64 KB (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.

25.7.30 `cpio_in_flags`

Flags for cpio commands that the SAP tools use for reading files from a volume.

Syntax: `cpio_in_flags = <flags>`

Default value: `-iuvB`

The option `B` causes a block size of 5KB to be used. You can also use a higher block size, such as 64 KB (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.

25.7.31 `dd_flags`

Defines the `dd` options required for writing to tape.

This parameter is required if you are working with raw devices. In order to access raw devices, the `dd` command is used. See [Raw Devices with BRBACKUP and BRRESTORE \[Page 34\]](#).

Syntax: `dd_flags = <flags>`

Default: `"obs=16k"`

If you define the default value, the data is written to tape in blocks of 16 KB, but you can also use a higher value, for example 64 KB.

25.7.32 `dd_in_flags`

Defines the `dd` options required for reading from tape.

This parameter is required if you are working with raw devices. To access raw devices, the `dd` command is used. See [Raw Devices with BRBACKUP and BRRESTORE \[Page 34\]](#).

25.7 Initialization Profile init<DBSID>.sap

Syntax: `dd_in_flags = <flags>`

Default value: `"ibs=16k"`

If you define the default value, the data is read from tape in blocks of 16 KB, but you can also use a higher value, for example 64 KB.

25.7.33 disk_copy_cmd

This parameter specifies the copy command to be used to copy files to local disks.

Syntax: `disk_copy_cmd = copy|dd|rman`

Default: `copy`

Possible values:

- `copy`: Data is copied by a `cp` (UNIX) or `copy` (Windows NT) command to disk.
- `dd`: Data is copied by `dd` command to disk.
- `rman`: Oracle Recovery Manager (RMAN) is called to copy data to disk. See [RMAN Backup with an External Backup Library \[Page 144\]](#).

25.7.34 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 `remote_host`.

Syntax: `dismount_cmd = <cmd>`

Default: none (for `backup_dev_type = tape_box|pipe_box` this parameter *must* be defined in the initialization profile).

The dismount command uses its own options:

```
dismount_cmd = "<dismount_cmd> $ $ [$]"
```

`<dismount_cmd>`: Command name

The command name might be, 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 addresses of the tape devices
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 Automatic Tape Changers \[Page 176\]](#)

[Mount / Dismount Command \[Page 177\]](#).

[mount_cmd \[Page 459\]](#)

[mount_par_file \[Page 460\]](#)

25.7.35 exec_parallel

This parameter specifies the maximum number of parallel copy processes.

Syntax: `exec_parallel = <number>`

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.

- 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` or `stage_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.

25.7.36 expir_period

This parameter specifies the expiration period for the tape volumes.

Syntax: `expir_period = <days>`

Default: 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 \[Page 95\]](#).

25.7.37 mount_cmd

25.7 Initialization Profile init<DBSID>.sap

This parameter defines the dismount command for the automatic demounting of a tape. 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 `remote_host`.

Syntax: `mount_cmd = <cmd>`

Default: none

If `backup_dev_type = tape_box` | `pipe_box` is defined, then this parameter **must** be defined in the initialization profile. For more information, see [backup_dev_type \[Page 447\]](#).

The mount command has its own options.

```
mount_cmd = "<mount_cmd> $ $ $ [$]"
```

`<mount_cmd>`: Command name

For example, `command name` might be `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 addresses of the tape devices
3. The tape names
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 Automatic Tape Changers \[Page 176\]](#)

[Mount / Dismount Command \[Page 177\]](#)

[dismount_cmd \[Page 458\]](#)

[mount_par_file \[Page 460\]](#)

25.7.38 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.

Syntax: `mount_par_file = <file>`

See also:

[-r|-parfile \[Page 373\]](#)

[Mount / Dismount Command \[Page 177\]](#)

25.7.39 new_db_home

This parameter must be set if you want to make a database copy on local disks using BRBACKUP ([backup_dev_type \[Page 447\]](#) = disk_copy|disk_standby). The name of the home directory of the database copy is defined in new_db_home.

Syntax: new_db_home = <dir>

Default: none

UNIX: <dir> is the new SAP database directory, that is, /oracle/<NEW_SID>

NT : <dir> is the new SAP database directory: <drive>:\oracle\<NEW_SID>

This directory must also contain the sapbackup directory.



With 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 \[Page 370\]](#)). If you do not specify a target drive, all files are copied to the directory defined in the parameter.

See also:

[Structure Retaining Database Copy \[Page 111\]](#)

[The SAP Utilities with Windows NT \[Page 508\]](#)

25.7.40 next_exclude

This parameter defines the database objects (that is, tables, indexes, and tablespaces) to be excluded from the [BRCONNECT next function \[Page 435\]](#).

Syntax:

```
next_exclude
= [<owner>.]<table>| [<owner>.]<index>| <tablespace>| (<object_list>) | all_part
```

Default: No exclusion, all SAP tables and indexes are processed

all_part means that SAP partitions (such as in Business Information Warehouse and Advanced Planner and Optimizer) are excluded from the adapting of extents.

You can use this parameter to exclude tables or indexes with exceptional space parameters from the BRCONNECT [next \[Page 435\]](#) function.

25.7.41 next_limit_count

This parameter defines the maximum number of next extents (MAXEXTENTS).

Input syntax: next_limit_count = <number>

Default: 0 (that is, no limit)

If specified, the [BRCONNECT next function \[Page 435\]](#) sets the MAXEXTENTS storage parameter for all processed tables and indexes to this value.

25.7.42 next_max_size

This parameter defines the upper limit for the next extent size, in KB, MB, or GB.

25.7 Initialization Profile init<DBSID>.sap

Syntax: `next_max_size = <size>`

where `<size>`: `<n>K` | `<n>M` | `<n>G` for the upper limit of the next extent size

Default: 2 GB - 5 * <database block size>

0 means unlimited next extent size

The [BRCONNECT next function \[Page 435\]](#) does **not** change the NEXTEXTENT storage parameter of tables or indexes to a value greater than the value specified in this parameter.



To set the upper limit for the next extent size to 1 MB, enter the following:
`next_max_size = 1M`

25.7.43  next_owner

This parameter defines the database owner of tables and indexes to be processed by the [BRCONNECT next function \[Page 435\]](#).

Syntax: `next_owner = <owner> | (<owner_list>)`

Default: `SAPR3/SAP<SID>` in a standard SAP database or all SAP owners in a multi-schema database

You can use this parameter to limit the `next` function to objects of selected SAP owners (that is, systems).

25.7.44  next_special

This parameter defines the special sizes of next extent for exceptional tables and indexes, in KB, MB, or GB. You can use this parameter to specify your own sizes for next extents and maximum number of extents for selected tables and indexes.

Syntax:

```
next_special =
[<owner>.]<table>:<size>[/<limit>], [<owner>.]<index>:<size>[/<limit>]
| (<special_list>) | all_sel:<size>[/<limit>]
```

where

`<size>`: `<n>K` | `<n>M` | `<n>G` special NEXTEXTENT size

`<limit>`: special MAXEXTENTS count

`all_sel`: sets NEXTEXTENT and MAXEXTENTS attributes to a certain value for **all** the database objects selected using the `-t` function option of [-f next \[Page 435\]](#) or the [next table \[Page 462\]](#) parameter. This option is provided for exceptional situations.

Default: According to table category

For more information, see [Internal Rules for Determining Next Extent Size \[Page 419\]](#).

25.7.45  next_table

This parameter defines the database objects to be processed by the [BRCONNECT next function \[Page 435\]](#).

Syntax:

```
next_table =
all|special| [<owner>.]<table>| [<owner>.]<index>|<tablespace>| (<object_list>)
```

Default: All objects of selected owners

all: All SAP tables and indexes

special: Only tables and indexes defined in the [next_special \[Page 462\]](#) parameter

25.7.46 orig_db_home

This parameter lets you rename the database file for the standby database and split mirror disk backup. The name of the home directory of the primary database is defined in `orig_db_home`.

Syntax: `orig_db_home = <dir>`

Default: none

UNIX: `<dir>` is the primary SAP database directory, that is, `/oracle/<SID>`

Windows: `<dir>` is the primary SAP database directory, that is, `<drive>:\oracle\<SID>`

Normally this parameter is only set in the profile on the standby or backup server.

See also:

[Standby Database \[Page 162\]](#)

[Split-Mirror Disk Backup \[Page 170\]](#)

25.7.47 parallel_instances

This parameter is only relevant if you are using Oracle Parallel Server (OPS).

The instances running in parallel to the dedicated database instance are defined by this parameter.

Syntax:

```
parallel_instances = <inst_descr>| (<inst_descr_list>)
```

Default: none.

`<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

`<inst_string>`: Oracle instance string for connect to the parallel instance



```
parallel_instances =
(C11_002:/oracle/C11@C11_002,C11_003:/oracle/C11@ C11_003)
```

25.7.48 primary_db

This parameter is only relevant for [Standby Database \[Page 162\]](#) and [Split-Mirror Disk Backup \[Page 170\]](#).

The connect string to the primary database instance is defined with this parameter so that BRBACKUP can log onto the primary host.

25.7 Initialization Profile init<DBSID>.sap

Syntax: `primary_db = <inst_string>`

Default value: None

`<inst_string>`: Instance string, or "SQL*Net database specification string," for the connect from the standby host to the primary (that is, production) database.



```
primary_db = C11
```

25.7.49  remote_host

Use this parameter to specify the name of the remote host, if you want to make a backup to a remote tape or disk.

Syntax: `remote_host = <host>`

Default: none

The parameters `remote_host` and `remote_user` replace the parameter `read_fifo_cmd`, which is no longer used from Release 4.5A onwards.

See also:

[remote_user \[Page 464\]](#)

[Backup to a Remote Disk \[Page 105\]](#)

25.7.50  remote_user

Use this parameter to specify the user for a remote host, if you want to make a backup to a remote tape or disk.

Syntax: `remote_user = <user>`

Default: none

If you use the ftp network to transfer data (see [stage_copy_cmd \[Page 470\]](#)) you can also specify the password which BRBACKUP uses to log on to the remote host.

```
remote_user = "<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 \[Page 105\]](#)

[remote_host \[Page 464\]](#)

25.7.51  restore_mode

This parameter is used by BRRESTORE to determine the scope of restore activity.

Syntax:

```
restore_mode =
all|all_data|full|incr|incr_only|incr_full|<tablespace>|<file_ID>|<f
ile_ID1>-<file_ID2>|(<object_list>]
```

Default: **all**

Possible values:

- **all**
Restore all tablespaces (without control files and redo log files)
- **all_data**
Restore all tablespaces that are not pure index tablespaces.
- **full**
Restore a complete backup (that is, complete or partial).
- **incr**
Restore an incremental backup with RMAN.
- **incr_only**
Restore changes to all files that were in the database at the time of the last full backup. See "Restoring Incremental Backups with Structural Changes" in [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).
- **incr_full**
Restore files that have been added to database since the last full backup. See "Restoring Incremental Backups with Structural Changes" in [Profile Parameters and BRBACKUP Command Options \[Page 380\]](#).
- **<tablespace>**
The files of the specified tablespaces are processed by the appropriate SAP utility.
- **<file_ID>**
The file with this file ID are restored. 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 restore all existing online redo log files.
- **<file_ID1>-<file-ID2>**
The files in this interval are restored.
- **<object_list>**
You can enter an object list, as described in [Initialization Profile init<DBSID>.sap \[Page 445\]](#). This list can also include the key word **all**. However, we recommend processing database files and non-database files separately.

25.7.52 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.

Syntax: **resync_cmd** = "<resync_cmd> [\$]"

Default: none

This command has its own option:

<resync_cmd>: Program or shell script for synchronizing the mirror disks.

25.7 Initialization Profile init<DBSID>.sap

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 Disk Backup \[Page 170\]](#)

[split_cmd \[Page 470\]](#)

25.7.53  rewind

This parameter defines the rewind command for your host operating system.

Syntax: `rewind = <cmd>`

Default: None

Used by the SAP tools 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.

For example, you might enter:

```
rewind = "mt -f $ rewind"
```



Rewind commands:

HP-UX: "mt -f \$ 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.

The SAP tools replace the \$ character with the address of the device used.

25.7.54  rewind_offline

This parameter defines the rewind/set offline command for your host operating system.

Syntax: `rewind_offline = <cmd>`

Default: The value of the [rewind \[Page 466\]](#) 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`).

You could make the following entry, for example:

```
rewind_offline = "mt -f $ offline"
```



Rewind commands:

HP-UX: "mt -f \$ 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.

The SAP tools replace the \$ character with the address of the device used.

See also:

[Backup with Automatic Tape Changers \[Page 176\]](#)

25.7.55 rman_channels

This parameter defines the number of parallel channels used by RMAN with an external backup library, that is, `backup_dev_type = rman_util`. For more information, see [backup_dev_type \[Page 447\]](#).

Syntax: `rman_channels = <number>`

Default: 1

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.56 rman_diskratio

This parameter defines the number of disk drives used for reading datafiles by RMAN with an external backup library, that is, `backup_dev_type = rman_util`.

Syntax: `rman_diskratio = <number>`

Default: 0 (use RMAN default value)

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.57 rman_duplex

This parameter defines the number of copies of each save set created by RMAN with an external backup library, that is, `backup_dev_type = rman_util`.

Syntax: `rman_duplex = 0|1|2|3|4`

Default: 0 (one copy, same as 1)

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.58 rman_filesperset

This parameter defines the number of database files stored in a save set by RMAN with an external backup library, that is, `backup_dev_type = rman_util`. For more information, see [backup_dev_type \[Page 447\]](#).

Syntax: `rman_filesperset = <number>`

25.7 Initialization Profile init<DBSID>.sap

Default: 0, which means one file per save set for non-incremental backups or all files in one save set for incremental backups

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.59  rman_kbytes

This parameter defines the maximum size in kilobytes of the backup pieces created by RMAN with an external backup library, that is, `backup_dev_type = rman_util`).

Syntax: `rman_kbytes = <number>`

Default: 0 (no limit)

You can also use this parameter to limit the size of a save set created by an RMAN incremental backup to disk.

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.60  rman_maxopenfiles

This parameter defines the maximum number of opened files by RMAN.

Syntax: `rman_maxopenfiles = <number>`

Default: 0 (no limit)

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.61  rman_parms

This parameter defines additional parameters that RMAN passes to an external backup library, that is, `backup_dev_type = rman_util`.

Syntax: `rman_parms = "<string>"`

Default: none

Ask the vendor of the external backup library if it requires you to set additional parameters.



```
rman_parms = "BLKSIZE=65536 ENV=(BACKUP_SERVER=HOSTNAME) "
```

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.62  rman_pool

This parameter defines the media pool in which the backups are stored by RMAN with an external backup library, that is, `backup_dev_type = rman_util`.

Syntax: `rman_pool = <number>`

Default: 0 (standard media pool)

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.63 rman_proxy

This parameter defines that the proxy copy functionality is to be used by RMAN with an external backup library, that is, `backup_dev_type = rman_util`.

Syntax: `rman_proxy = no|yes|only`

Default: no

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.64 rman_readrate

This parameter defines the maximum number of buffers per second that will be read by RMAN for each datafile.

Syntax: `rman_readrate = <number>`

Default = 0 (no limit)

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.65 rman_setsize

This parameter defines the maximum size of save sets created by RMAN with an external backup library, that is, `backup_dev_type = rman_util`.

Syntax: `rman_setsize = <number>`

Default: 0 (no limit)

See also:

[RMAN Backup with an External Backup Library \[Page 144\]](#)

Oracle documentation

25.7.66 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.

25.7 Initialization Profile init<DBSID>.sap

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.

Syntax: `saveset_members = 1|2|3|4|tsp|all`

Default: 1 (no file multiplexing)

1, 2, 3, 4: Number of files in a save set

`tsp`: each save set contains all files of a tablespace

`all`: only one save set with all database files is created

See also:

[RMAN Save-Set Grouping \[Page 151\]](#)

25.7.67  split_cmd

This parameter is only relevant when using the [split-mirror \[Page 170\]](#) configuration.

This parameter contains a program or shell script (with or without options) called by BRBACKUP to split the mirror disks.

Syntax: `split_cmd = <cmd>`

Default: none

This command has its own options:

```
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.

25.7.68  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.

Syntax: `stage_copy_cmd = rcp|ftp`

Default: `rcp`

`rcp` means that the operating system remote copy program is used

`ftp` means that SAP implementation of file transfer protocol (`sapftp`) is used.

An incremental backup to a remote disk is always made with `ftp`. In this case, this parameter is ignored.

See also:

[Backup to a Remote Disk \[Page 105\]](#)

25.7.69 stage_db_home

This parameter defines the name of the home directory for the database copy. You must set this parameter if you want to make a database copy on remote disk using BRBACKUP, that is, `backup_dev_type = stage_copy|stage_standby`.

Syntax: `stage_db_home = <dir>`

Default: value of [new_db_home parameter \[Page 461\]](#)

UNIX: `<dir>` is the new SAP database directory, `/oracle/<NEW_SID>`

Windows NT: `<dir>` is the new SAP database directory, `<drive>:\oracle\<NEW_SID>`



With 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 \[Page 370\]](#)). If you do not specify a target drive, all files are copied to the directory defined in the parameter.

See also:

[Structure-Retaining Database Copy \[Page 111\]](#)

[The SAP Utilities with Windows NT \[Page 508\]](#)

25.7.70 stage_root_dir

This parameter identifies the directories used by BRBACKUP to back up database backups to a remote disk. In exceptions (for example, no definition of the parameter `archive_stage_dir`), BRARCHIVE also uses the directories defined in this parameter if remote disk archiving is required. If you enter more than one directory, you must enclose the names in parentheses and separate them by commas.

Syntax: `stage_root_dir = <dir>|(<dir_list>)`

Default: The directory defined in the parameter [backup_root_dir \[Page 450\]](#).

Required value: Directories 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 \[Page 105\]](#)

25.7.71 stats_bucket_count

This parameter defines the number of buckets for updating statistics with histograms for the [BRCONNECT update statistics function \[Page 436\]](#).

Syntax: `stats_bucket_count = <number>`

Default: 75

25.7.72 stats_change_threshold

This parameter defines the threshold for the percentage of inserted or deleted rows causing an update in optimizer statistics for the [BRCONNECT update statistics function \[Page 436\]](#).

25.7 Initialization Profile init<DBSID>.sap

Syntax: `stats_change_threshold = <number>`

Default: 50

The value must be greater than zero.



25% means that statistics are collected for a table if either of the following is true:

- Current row count is **greater than or equal to** (100 + 25) % of the old row count
- Current row count is **less than or equal to** 100/(100 + 25) % of the old row count

25.7.73  stats_dbms_stats

This parameter defines how the DBMS_STATS package is used to update statistics with BRCONNECT.

Syntax: `stats_dbms_stats = [ALL:<X>:<degree> | [<owner>.]<table>:<X>:<degree> | <entry list> | info_cubes:<X>:<degree>`

If **x** = R: activates row sampling

If **x** = B: activates block sampling

<degree>: degree of parallelism for the collection of statistics

info_cubes: only processes InfoCube tables with DBMS_STATS package

The optional entry `ALL:R | B:<degree>` activates the DBMS_STATS package for all selected tables and defines the standard sampling type and degree of parallelism. Without this entry, only the specified tables are processed.

The sampling type and degree of parallelism that you enter for the specified tables overrides the default sampling type and degree of parallelism (as defined for a table in Oracle dictionary).



- `stats_dbms_stats = ALL:R:1`
This parameter activates the DBMS_STATS package with row sampling for all selected tables, without parallelism.
- `stats_dbms_stats = (ALL:R:2, ATAB:B:3, RFBLG:B:4)`
This parameter activates the DBMS_STATS package for:
 - ATAB table, with block sampling and parallelism degree 3
 - RFBLG table with block sampling and parallelism degree 4
 - All other selected tables with row sampling and parallelism degree 2
- `stats_dbms_stats = (ATAB:R:3, RFBLG:R:4)`
This parameter activates the DBMS_STATS package for:
 - ATAB table, with row sampling and parallelism degree 3
 - RFBLG table with row sampling and parallelism degree 4
 All other selected tables are processed with the ANALYZE statement

For more information on DBMS_STATS, see [Update Statistics \[Page 71\]](#)

25.7.74  stats_exclude

This parameter defines the database objects (that is, tables, indexes, and tablespaces) to be excluded from the [BRCONNECT update statistics function \[Page 436\]](#). You can use this parameter to exclude tables or indexes with exceptional statistics handling from update statistics.

Syntax:

```
stats_exclude =
[<owner>.]<table><[<owner>.]<index>|<tablespace>|([object_list])|info_cubes
```

Default: No exclusion, all SAP tables and indexes are processed

info_cubes: Exclude all InfoCube tables from updating statistics.

25.7.75 stats_info_cubes

This parameter defines which tables are treated as InfoCube tables by BRCONNECT. For more information, see [Update Statistics for InfoCube Tables \[Page 425\]](#).

Syntax: stats_info_cubes =
null|default|[<owner>.]<table>|[<owner>.]<prefix>*|(<table_list>)

null: suppresses special handling of InfoCube tables.

default: includes the default table list of InfoCube tables specified in [Update Statistics for InfoCube Tables \[Page 425\]](#)



```
stats_info_cubes = (default,ABC*,XYZ*)
```

In this example, table ABC and all tables starting with XYZ are treated as InfoCube tables, in addition to default tables.

25.7.76 stats_limit_time

This parameter defines the processing time limit in minutes for updating statistics with the [BRCONNECT update statistics function \[Page 436\]](#).

You can use this parameter to terminate long-running update statistics jobs after a period of time. The processing terminates after statistics have been collected for the current table or index (this is the "soft limit"). However, if you set the option `-f limit` in the [update statistics function \[Page 436\]](#), processing terminates immediately (this is the "hard limit").

Syntax: stats_limit_time = <minutes>

Default: 0, that is, no limit

25.7.77 stats_method

This parameter defines the method to be applied to tables that are not specified in the DBSTATC control table for the [BRCONNECT update statistics function \[Page 436\]](#).

Syntax: stats_method = E|EH|E|EI|EX|C|CH|CI|CX|E=|C=|H=|I=|X|+H|+I

Default: Internal rules determine the update statistics method, as with [stats_sample_size \[Page 474\]](#)

Possible values:

- E: Estimate

25.7 Initialization Profile init<DBSID>.sap

- **EH**: Estimate with histograms
- **EI**: Estimate with index validation
- **EX**: Estimate with histograms and index validation
- **C**: Compute
- **CH**: Compute with histograms
- **CI**: Compute with index validation
- **CX**: Compute with histograms and index validation
- **E=**: Force estimate for all tables, including tables in DBSTATC control table (option -f **method** must be set)
- **C=**: Force compute for all tables, including tables in DBSTATC control table (option -f **method** must be set)
- **=H**: Force creation of histograms for all tables, including tables in DBSTATC control table (option -f **method** must be set)
- **=I**: Force index validation for all tables, including tables in DBSTATC control table (option -f **method** must be set)
- **=X**: Force creation of histograms and index validation for all tables, including tables in DBSTATC control table (option -f **method** must be set)
- **+H**: Force creation of histograms for all tables, including tables in DBSTATC control table in addition in addition to index validation, if defined there (option -f **method** must be set)
- **+I**: Force estimate with index validation for all tables, including tables in DBSTATC control table in addition to creating histograms, if defined there (option -f **method** must be set)

25.7.78  stats_owner

This parameter defines the database owner of tables and indexes for the [BRCONNECT update statistics function \[Page 436\]](#). You can use this parameter to limit processing to objects of selected SAP owners (that is, systems).

Syntax: **stats_owner** = <owner> | (<owner_list>)

Default: **SAPR3/SAP<SID>** in a standard SAP database or all SAP owners in a multi-schema database

25.7.79  stats_parallel_degree

This parameter defines the number of parallel threads for updating statistics with the [BRCONNECT update statistics function \[Page 436\]](#). For example, you can set this parameter to the number of CPUs to speed up update statistics.

Syntax: **stats_parallel_degree** = <number>

Default: 1

25.7.80  stats_sample_size

This parameter defines the sample size with the [BRCONNECT update statistics function \[Page 436\]](#) using method E for tables that are **not** specified in the DBSTATC control table.

Syntax: `stats_sample_size = P<p>|R<r>`

Default: [Internal rules \[Page 424\]](#) determine the sample size, as with [stats_method \[Page 473\]](#)

P<p>: percentage of rows

R<r>: thousands of rows

25.7.81 stats_table

This parameter defines the database objects to be processed by the [BRCONNECT update statistics function \[Page 436\]](#) [\[Page 436\]](#). You can use this parameter to restrict update statistics to selected objects.

Syntax:

```
stats_table =
all|all_ind|missing|dbstatc_tab|dbstatc_mon|dbstatc_mona| [<owner>.]<
table>| [<owner>.]<index>|<tablespace>| (<object_list>) |info_cubes
```

Default: All objects of selected owners

- **all**: All SAP tables and indexes
- **all_ind**: All indexes only. For example, you can use this to create space statistics for all indexes.
- **missing**: Only tables and indexes with missing statistics
- **dbstatc_tab**: Only tables specified in the DBSTATC control table
- **dbstatc_mon**: Only tables specified in the DBSTATC control table that are relevant for the application monitor
- **dbstatc_mona**: Only application tables specified in the DBSTATC control table that are relevant for the application monitor
- **info_cubes**: Only InfoCube tables.

25.7.82 tape_address

This parameter is used by the SAP tools to identify the device addresses to 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`.

Syntax: `tape_address = <dev>| (<dev_list>)`

Default: 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.



Sample specification of backup device:

25.7 Initialization Profile init<DBSID>.sap

```
HP-UX, Solaris: /dev/rmt/0mn; AIX: /dev/rmt0.1; TRU64:
/dev/nrmt0h; Reliant: /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 \[Page 105\]](#)). 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.

BRRESTORE can also use several backup devices in parallel. See [Restoring Files \[Page 396\]](#).

The number of device addresses in `tape_address` must agree with the number of device addresses in `tape_address_new`.

25.7.83 `tape_address_arch`

BRARCHIVE uses this parameter to identify device addresses for writing to a volume (tape).

Syntax: `tape_address_arch = <dev> | (<dev_list>)`

Default: value of [tape_address \[Page 475\]](#)

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 \[Page 475\]](#).

The number of device addresses in `tape_address_arch` must agree with the number of device addresses in `tape_address_arch_rew`.

25.7.84 `tape_address_ctl`

This parameter is used by SAP tools 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, that is, `backup_dev_type = tape_box|pipe_box`. For more information, see [backup_dev_type \[Page 447\]](#).

Syntax:

`tape_address_ctl = <dev> | (<dev_list>)`

Default: value of [tape_address_rew \[Page 477\]](#)

<dev>: Control driver address

The number of control driver addresses in `tape_address_ctl` must agree with the number of device addresses in `tape_address`.

See also:

[Backup with Automatic Tape Changers \[Page 176\]](#)

25.7.85 `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`).

Syntax: `tape_address_ctl_arch = <dev> | (<dev_list>)`

Default: value of [tape_address_rew_arch \[Page 477\]](#)

If this parameter is not set, BRARCHIVE uses the drivers specified in the parameter [tape_address_ctl](#) [Page 476].

<dev>: Control driver address

The number of `tape_address_ctl_arch` must agree with the number specified in [tape_address_arch](#) [Page 476].

See also:

[Backup with Automatic Tape Changers](#) [Page 176]

25.7.86 `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`.

Syntax: `tape_address_rew = <dev> | (<dev_list>)`

Default: 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.



Sample specification of backup device:

HP-UX, Solaris: `/dev/rmt/0m`; AIX: `/dev/rmt0`; TRU64:
`/dev/rmt0h`; Reliant: `/dev/ios0/rstape005`; NT: `/dev/mt0`

Also see [tape_address](#) [Page 475].

The number of device addresses in `tape_address_rew` must agree with the number of device addresses in [tape_address](#) [Page 475].

25.7.87 `tape_address_rew_arch`

BRARCHIVE uses this parameter to identify device addresses for writing to a volume (tape).

Syntax: `tape_address_rew_arch = <dev> | (<dev_list>)`

Default: value of [tape_address_rew](#) [Page 477]

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 477].

See also [tape_address_arch](#) [Page 476].

The number of device addresses in `tape_address_rew_arch` must agree with the number of device addresses in parameter [tape_address_arch](#) [Page 476].

25.7.88 `tape_copy_cmd`

Command used to copy database files (not to raw devices) and non-database files from disk to tape.

Syntax: `tape_copy_cmd = cpio | dd | cpio_gnu | rman | rman_dd`

Default: `cpio`

25.7 Initialization Profile init<DBSID>.sap

cpio: Data is copied by cpio command

dd: Data is copied by dd command

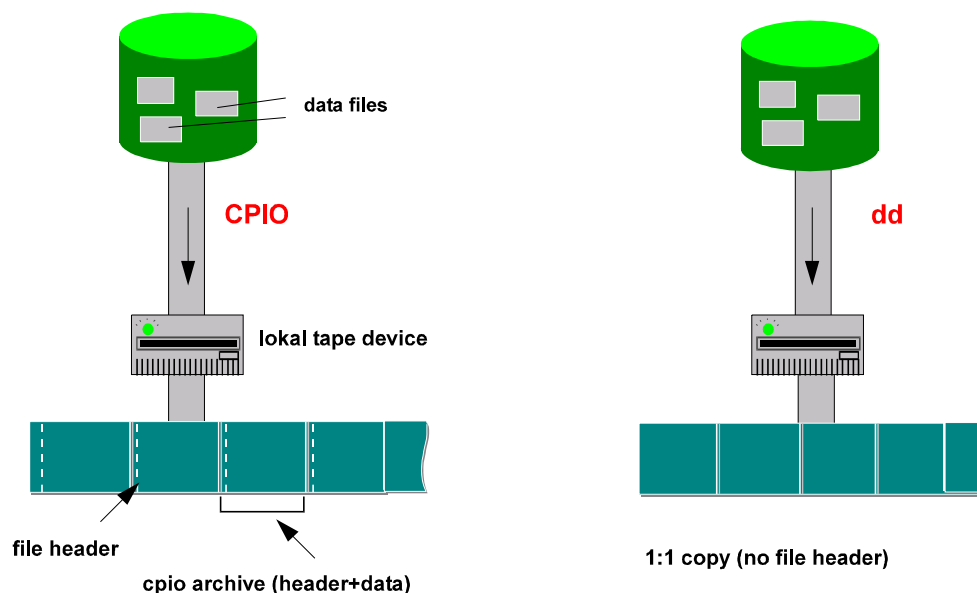
cpio_gnu: GNU command syntax command used to copy files. On some platforms, GNU cpio offers better performance.

rman: Backup with Oracle Recovery Manager. Data is copied by SAP backup library directly to tape.

rman_dd: Backup with Oracle Recovery Manager. Data is copied by SAP backup library piped through dd command to tape.

For more information on **rman** and **rman_dd**, see [RMAN Backup with the SAP Backup Library \[Page 142\]](#).

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 back up the database can improve 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"

Windows NT:
`dd_flags = "bs=xk"`
`dd_in_flags = "bs=xk"`
 where `x` >=16
 The `dd` options `obs` and `ibs` are not supported on Windows NT.

25.7.89 `tape_pos_cmd`

Command for positioning the read head on the tape.

Syntax: `tape_pos_cmd = <cmd>`

Default: 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 $"`



Sample positioning commands:
 HP-UX: `"mt -t $ fsf $"`; AIX: `"tctl -f $ fsf $"`; TRU64, Reliant,
 Solaris, NT: `"mt -f $ fsf $"`;

Make sure that you do not forget the double quotes when you enter parameters composed of several values. The SAP tool 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.

25.7.90 `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.

Syntax: `tape_size = <size>K|<size>M|<size>G`

Default: 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 \[Ext.\]](#).

25.7.91 `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.

Syntax: `tape_size_arch = <size>K|<size>M|<size>G`

Default: the value set in [tape_size \[Page 479\]](#)

25.7 Initialization Profile init<DBSID>.sap**25.7.92  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.

Syntax: `tape_use_count = <number>`

Default value: 100

25.7.93  uncompress_cmd

Provides the command used to decompress files that were backed up with software compression.

Syntax: `uncompress_cmd = <cmd>`

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 $ > $"
```

Be sure to follow the syntax rules. In particular, the double quotes " are essential.

25.7.94  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.

Syntax: `util_par_file = <file>`

Default: 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 the directory:

```
<ORACLE_HOME>/dbs (UNIX)
```

```
<ORACLE_HOME>\database (Windows NT)
```

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`.

For more information, see [External Backup Programs \[Page 180\]](#).

25.7.95  volume_archive

This parameter is used by BRARCHIVE to identify the volume set to be used for the archive of the offline redo log files.

Syntax: `volume_archive = <vol> | (<vol_list>)`

Default: 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.

When BRARCHIVE is started, the automatic volume management checks all the volumes in the sequence of their names in `volume_archive` (only volumes whose expiration period has expired are suggested for archiving). The volumes are 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 [Selecting Volumes Automatically \[Page 100\]](#).

The expiration period can be configured using parameter `expir_period`. The automatic volume management ensures that only free volumes are offered for backup, for example, those whose expiration period has expired. As a result, you 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% – for example, specifying 18 or more volume names in this case.

SCRATCH

To deactivate the automatic volume management, use the command `volume_archive = SCRATCH`. In this case, you can mount any volume whose expiration period has expired. This name is then also recorded during the archive.

25.7.96 volume_backup

This parameter is used by BRBACKUP to identify the volume set to be used for the backup of the database or non-database files.

Syntax: `volume_backup = <vol>|<vol_list>`

Default: 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.

When BRBACKUP is started, the automatic volume management checks all the volumes in the sequence of their names in `volume_backup` (only volumes whose expiration period has expired are suggested for backup). The volumes are suggested cyclically.

25.8 Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT



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 prompts you to mount the third selected volume at the appropriate time.

See [Selecting Volumes Automatically \[Page 100\]](#).

The expiration period can be configured using parameter `expir_period`. The automatic volume management ensures that only free volumes are offered for backup, for example, those whose expiration period has expired. As a result, you 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%, for example, 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 is then also recorded during the backup.

25.8 Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT

Logs are created when calling the SAP programs BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT. These supply information about database operations and are useful for problem 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).



Be sure to include logs if you need to create a problem message, as this simplifies error analysis.

For more information, see:

- [Log Types \[Page 482\]](#)
- [BRBACKUP Logs \[Page 377\]](#)
- [BRARCHIVE Logs \[Page 392\]](#)
- [BRRESTORE Logs \[Page 406\]](#)
- [BRCONNECT Logs \[Page 443\]](#)

25.8.1 Log Types

The following types of logs are written:

- [File System Logs \[Page 483\]](#)

25.8 Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT

- [Database Logs \[Page 483\]](#) in tables SDBAH and SDBAD

25.8.2 File System Logs

BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT write the following file system logs:

- A detail log that is generated each time a process is started.
- A summary log that is appended after every run.

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:

- `<SAPDATA_HOME>/sapbackup` – for BRBACKUP, BRRESTORE
- `<SAPDATA_HOME>/saparch` – for BRARCHIVE
- `<SAPDATA_HOME>/sapcheck` – for BRCONNECT

25.8.3 Database Logs for BRBACKUP, BRARCHIVE, and BRCONNECT

The database logs of BRBACKUP, BRARCHIVE, and BRCONNECT are stored in tables SDBAH and SDBAD. SAPDBA also uses these tables. SAPDBA saves information from the main log `reorg<DBSID>.log` in these database tables. For more information, see [Central Log reorg<DBSID>.log \[Page 207\]](#).



Do **not** manually change tables SDBAH and SDBAD. Only the appropriate SAP tools and the SAP System should append or change these tables.

This section does not describe these tables in detail, but instead gives an overview of which information BRBACKUP, BRARCHIVE, and BRCONNECT stores in which tables.

SDBAH

This table contains information that refers to the backup as a whole:

- Starting time of the backup or BRCONNECT processing
- End time of the backup or BRCONNECT processing
- BRBACKUP/BRARCHIVE/BRCONNECT return code
- BRBACKUP/BRARCHIVE/BRCONNECT action ID (encoded timestamp of the file system log names)
- BRBACKUP/BRARCHIVE/BRCONNECT 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

25.8 Logs for BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT

- 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.

For BRCONNECT, `SDBAD` tables contain information on the total number of objects processed by BRCONNECT.

25.8.4 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 \[Page 372\]](#) or [-w|-verify \[Page 376\]](#) (for BRBACKUP), [-w|-verify \[Page 391\]](#) (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 BRBACKUP 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 backup devices. When optimization by time was not possible BRBACKUP attempts to equally distribute the data volume.

Log Supplements When Using: `-o time`

25.9 Messages and Return Codes for the BR Tools

During the processing, SAP tools issue 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 need 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).

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.

25.9 Messages and Return Codes for the BR Tools

Messages

The BRBACKUP, BRARCHIVE, BRRESTORE, BRCONNECT tools 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 tools issue a group of messages, which will enable you to precisely analyze the error.

Return Codes

The following return codes are possible:

0	Successful
1	Warnings – all files were processed (for example, backed up or restored)

25.9 Messages and Return Codes for the BR Tools

2	Canceled during the initialization phase by a user or other signal
3	Errors occurred during the initialization phase, processing was not started
4	Canceled by a user or other signal during processing
5	Started, but not completed because errors occurred during processing
6	Internal termination

26 Database Recovery with SQLPLUS

Use

Although [SAPDBA recovery functions \[Page 321\]](#) can in most cases be used to recover a failed database, there are exceptions to this. Therefore, this section describes how to recover the Oracle database system using Oracle functions.

The information here helps you with database recovery after an error in one of the following database components:

- Data files
- Online redo log files
- Control files



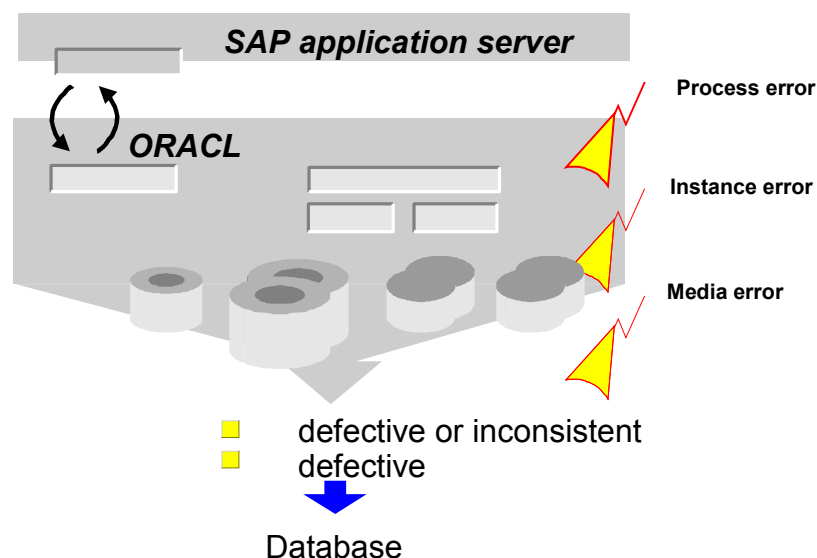
An incorrectly performed recovery might lead to irreparable loss of data in your database. We recommend you to always discuss your recovery plan with an experienced co-worker, your SAP consultant, or an Oracle specialist before you start a recovery.

26.1 Types of Database Errors

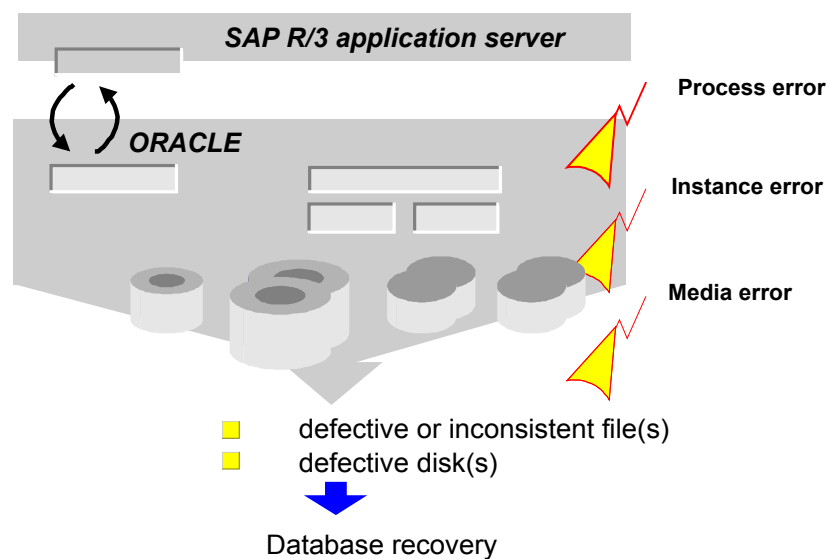
Use

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.



26.1 Types of Database Errors



Errors can generally be divided into five categories:

- User errors, see [Recovery after User Errors \[Page 489\]](#)
- Statement errors, see [Recovery after Statement Errors \[Page 489\]](#)
- Process errors, see [Recovery after Process Errors \[Page 489\]](#)
- Instance errors, see [Recovery after an Instance Error \[Page 490\]](#)
 - Media errors, see [Recovery after Media Errors \[Page 490\]](#)

[Error Analysis \[Page 488\]](#)

26.1.1 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: `<SAPDATA_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:

`<SAPDATA_HOME>/saptrace/background/<process>_<number>.trc`

Trace files of the user processes:

`<SAPDATA_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).

26.2 Recovery after User Errors

A user error occurs when a user deletes or falsifies data (for example, 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 Dictionary or the ABAP 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.

26.3 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 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.

26.4 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 continue normally.

26.5 Recovery after an Instance Error

The database administrator does not have to intervene to perform a recovery after a process error.

26.5 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.

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.

26.6 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, be sure to contact your SAP consultant or an Oracle specialist. Do **not** attempt a recovery if you are not confident.

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

26.7 Recovery: One Control File Missing

defects in the data files of the database are involved, SAPDBA can often perform the database recovery automatically.

See [SAPDBA Database Recovery \[Page 321\]](#).

This section is intended to provide an overview of the recovery process using Oracle utilities. Use Oracle SQLPLUS. The sections below contain examples of the command line operations with SQLPLUS.

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 (for example, if the archiving directory for the offline redo log files is lost or full).

See also:

- [Recovery: One Control File Missing \[Page 491\]](#)
- [Recovery: All Control Files Missing \[Page 492\]](#)
- [Recovery: Current Online Redo Log Missing \[Page 495\]](#)
- [Recovery: One Inactive Online Redo Log Missing \[Page 496\]](#)
- [Recovery: User Tablespace Missing \[Page 497\]](#)
- [Recovery: SYSTEM Tablespace Missing \[Page 499\]](#)
- [Recovery: Index Tablespace Missing \[Page 500\]](#)
- [Recovery: Errors During the Archiving of Online Redo Logs \[Page 501\]](#)
- [Partial Recovery \[Page 502\]](#)
- [Actions after a Partial Recovery \[Page 504\]](#)
- [Automating the Recovery \[Page 506\]](#)
- [Updating the Control File \[Page 506\]](#)

26.7 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

26.8 Recovery: All Control Files Missing

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 SQLPLUS.
- Copy an existing control file to the desired storage location at operating system level (see parameter `control_files` in the `init.ora` profile).
- Start SQLPLUS 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 SQLPLUS.
- Delete the missing file from parameter `control_files` in profile `init.ora` (default: `<ORACLE_HOME>/dbs/init<DBSID>.ora`).
- Start SQLPLUS 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 \[Page 45\]](#)).

26.8 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.

26.8 Recovery: All Control Files Missing

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.

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: `<SAPDATA_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 SQLPLUS 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 \[Page 506\]](#).

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 \[Page 506\]](#).

7. Start recovery with the following SQLPLUS 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.

26.8 Recovery: All Control Files Missing

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 SQLPLUS 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.

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 \[Page 504\]](#).

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 SQLPLUS 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 SQLPLUS 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):

26.9 Recovery: Current Online Redo Log Missing

```

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.

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 SQLPLUS 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.

26.9 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.

26.10 Recovery: One Inactive Online Redo Log Missing

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 SQLPLUS 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 SQLPLUS 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 \[Page 502\]](#).

26.10 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, see 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 (for example, 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.

26.11 Recovery: User Tablespace Missing

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 SQLPLUS 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 SQLPLUS 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;
```

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.

26.11 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.

26.11 Recovery: User Tablespace Missing



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, see the Oracle documentation.

Proceed as follows during recovery:

1. Shut down the database system, if it is active, with the following SQLPLUS 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.

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 \[Page 398\]](#)).

5. Mount the database. Enter the following SQLPLUS 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:

26.12 Recovery: SYSTEM Tablespace Missing

```
alter database rename file '<file name>' to '<file name>';
```

See [Updating the Control File \[Page 506\]](#).

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 SQLPLUS command:

```
alter database datafile '<file name>' online;
```

See [Updating the Control File \[Page 506\]](#).

8. Use the following SQLPLUS 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 SQLPLUS 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 \[Page 321\]](#)).

26.12 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 SQLPLUS command to shut it down:

```
shutdown abort
```

26.13 Recovery: Index Tablespace Missing

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 \[Page 398\]](#)).

5. Mount the database. Enter the following SQLPLUS 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 \[Page 506\]](#).
7. The database administrator can automate the recovery using the `autorecovery` option. See [Automating the Recovery \[Page 506\]](#). Also read the corresponding Oracle documentation.
8. Use the following SQLPLUS 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 SQLPLUS command to open the database again:

```
alter database open;
```



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 \[Page 321\]](#)).

26.13 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 SQLPLUS 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 SQLPLUS commands:

26.14 Recovery: Errors During the Archiving of Online Redo Logs

```
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 SQLPLUS 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, see 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.

26.14 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 the following 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 SQLPLUS commands:

```
archive log list
```

26.15 Partial Recovery

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 SQLPLUS 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 [Setting Up Archiving \[Page 43\]](#).

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 SQLPLUS commands:

```
archive log stop
archive log start '<new path>'
```

`<new path>` is the name of the “directory” that will be used for archiving. 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 [Setting Up Archiving \[Page 43\]](#)). The database must remain active when you stop and restart the ARCH process.

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.

26.15 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, see 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 SQLPLUS 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 \[Page 398\]](#)).

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 SQLPLUS 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 \[Page 506\]](#).

7. You may be able to automate the recovery.

See [Automating the Recovery \[Page 506\]](#).

26.15 Partial Recovery

8. Use the following SQLPLUS 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 SQLPLUS 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.

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 \[Page 504\]](#).

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 SQLPLUS 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 \[Page 504\]](#).

26.15.1 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
```

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.

26.16 Automating the Recovery

26.16 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 SQLPLUS 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 [Setting Up Archiving \[Page 43\]](#)).

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.

26.17 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 SQLPLUS commands:

```
connect internal
startup mount
```

To update the path of data files in the control file, use the SQLPLUS 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. However, note 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 SQLPLUS 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 SQLPLUS commands:

```
connect internal
```

```
startup mount
```

```
alter database datafile '<complete file name>' online;
```

27 The SAP Utilities with Windows NT

Use

This section describes special features of SAPDBA, BRBACKUP, BRARCHIVE, and BRRESTORE with Windows NT.

For more information, see:

- [SAP Conventions \(Windows NT\) \[Page 508\]](#)
- [Backup Strategy \(Windows NT\) \[Page 511\]](#)

27.1 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 \(Windows NT\) \[Page 53\]](#)

[Directory Structure \(Windows NT\) \[Page 60\]](#)

[Naming Conventions for Files \(Windows NT\) \[Page 508\]](#)

[Executables \[Page 509\]](#)

[Starting the SAP Utility Programs \[Page 509\]](#)

[Reorganization \(Windows NT\) \[Page 510\]](#)

[Database Analysis \[Page 510\]](#)

[Restrictions on SAPDBA \(Windows NT\) \[Page 511\]](#)

27.1.1 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\pool1d_1\pool1d.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.

27.1.2 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 `mkzip` command is required if you want `BRBACKUP` or `BRARCHIVE` to save files with software compression (option `-k yes`).

This program is not used anymore to determine the compression rate of tape devices with hardware compression (option `-k only`). Instead, `BRTOOLS` is called, which uses the SAP compression library.

Oracle Programs

`SVRMGR30`, `EXP80`, `IMP80`, `SQLLDR80`

See also:

[Starting the SAP Utility Programs \[Page 509\]](#)

27.1.3 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 local groups `ORA_<DBSID>_DBA` and `ORA_<DBSID>_OPER` (or `ORA_DBA` and `ORA_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 `SQLPLUS` you no longer have 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>
```

27.1 SAP Conventions (Windows NT)

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 SQLPLUS as the user `internal`. Give the following authorizations to the user `system`:
`grant sysdba to system;`
`grant sysoper to system;`
 - Give a new password to the user `system` (optional):
`alter user system identified by <password>;`

Now you can call BRBACKUP as usual:
`brbackup -u system/<password>`

27.1.4 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 \[Page 36\]](#) still apply, however.

You can start the [reorganization in the background \[Page 283\]](#). 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

27.1.5 Database Analysis

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 `brconnect -f 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 `brconnect -f startup` or `brconnect -f shutdown`. For more information, see [SAPDBA Command Options \[Page 344\]](#).

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>
brconnect -u|-c -f 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
brconnect -u|-c -f check
```

To check any of the background statements that have been set, enter `at` in the command line.

27.1.6 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

27.2 Backup Strategy (Windows NT)

The following backup programs are available:

- [NTBackup \[Page 512\]](#) (Microsoft)
- [BRBACKUP \[Page 512\]](#) (SAP)
- [BARCHIVE \[Page 512\]](#) (SAP)

This section describes 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.



We recommend deleting the environment variable `ORACLE_HOME` for the database upgrade to Oracle 8.1.X. However, in this case the backup with older versions of BRBACKUP or BRARCHIVE terminates with the following error message:
BR152E Environment variable ORACLE_HOME is not set

27.2 Backup Strategy (Windows NT)

The problem occurs in particular with Business Warehouse 2.0, where Oracle 8.1 is used as a default.

For more information about solving this problem, see SAP Note 309322.

27.2.1 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 the [SAPDBA database restore and recovery functions \[Page 321\]](#) if the backup was created using NTBackup.

27.2.2 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, using ARCHIVELOG mode. For more information, see [Setting Up Archiving \[Page 43\]](#).

Offline redo log files can be archived to tape using BRARCHIVE. You can restore a complete backup using BRRESTORE. In a [recovery with SAPDBA \[Page 321\]](#), SAPDBA automatically restores the backup.

Prerequisites

BRBACKUP enables parallel backups to tape and disks. Some parameters in the initialization profile `init<DBSID>.sap` are specific to the operating system. (For example: `rewind_offline = "mt -f $ offline"`).

For more information about authorization checks, see [Executables \[Page 509\]](#).

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 \[Page 510\]](#)

27.2.3 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.

For more information, see [External Backup Programs \[Page 180\]](#).

27.2.4 Structure-Retaining Database Copy or Restore on Windows NT

Use

If the database is on several drives on Windows NT, problems occur as follows:

- Database copy with retained structure, that is, `backup_dev_type = disk_copy | disk_standby | stage_copy | stage_standby`
BRBACKUP attempts to copy all databases onto the drive which is defined in the `init<DBSID>.sap` parameter `new_db_home`.
- Restore on another computer with changed database name:
BRRESTORE attempts to reload all databases to the drive which is defined in the environment variable `SAPDATA_HOME`.

In both cases the distribution of the database files to several drives is not maintained.

For more information about solving this problem, see SAP Note 122363.

27.2.5 Offline Backup with Oracle Fail Safe for Cluster Systems

Use

Offline backup with Microsoft Cluster Server (MSCS) is not possible using the DBA Planning Calendar in the Computer Center Management System (CCMS), that is, with transaction DB13. The problem is that BRBACKUP calls SVRMGR to stop the database, but SQLPLUS is not allowed in the MSCS environment.

For more information about using Oracle Fail Safe to stop the database, see SAP Note 378648.