# Topic 1: Database Architecture and ASM

## ASM Shutdow abort process:

The Oracle ASM instance immediately shuts down without the orderly dismount of disk groups. This causes recovery to occur upon the next Oracle ASM startup.

If any database instance is connected to the Oracle ASM instance, then the database instance aborts.

## ASM support file types:

Table 7-1 File types supported by Oracle ASM

| **File Type** | **Default Templates** |
| --- | --- |
| Control files | CONTROLFILE |
| Data files | DATAFILE |
| Redo log files | ONLINELOG |
| Archive log files | ARCHIVELOG |
| Temporary files | TEMPFILE |
| Data file backup pieces | BACKUPSET |
| Data file incremental backup pieces | BACKUPSET |
| Archive log backup piece | BACKUPSET |
| Data file copy | DATAFILE |
| Persistent initialization parameter file (SPFILE) | PARAMETERFILE |
| Flashback logs | FLASHBACK |
| Change tracking file | CHANGETRACKING |
| Data Pump dumpset | DUMPSET |
| Automatically generated control file backup | AUTOBACKUP |
| Cross-platform transportable data files | XTRANSPORT |
| Flash file | FLASHFILE |
| Oracle ASM Persistent initialization parameter file (SPFILE) | ASMPARAMETERFILE |
| Oracle ASM Persistent initialization parameter file (SPFILE) backup | ASMPARAMETERFILEBACKUP |
| Oracle Cluster Registry file | OCRFILE |
| Oracle ASM Dynamic Volume Manager volumes | n/a |

Oracle ASM cannot directly support some administrative file types on disk groups. These include trace files, audit files, alert logs, export files, tar files, and core files.

## Alter diskgroup and V$ASM\_OPERATION

"Add directory" doesn't use metadata for recording status because "IT ISNT AN OPERATION"

## Communication between ASM instance and database instance

Through ASMB: manage storage and provide statistics.

## Maximize speed of adding/dropping disks from diskgroup

The most consumed time in adding/dropping action is REBALANCE DISK so define the POWER option in adding/dropping command or config ASM\_POWER\_LIMIT for more ARBn processes

## ASM Instance Startup

When you run the STARTUP command, rather than trying to mount and open a database, this command attempts to mount the disk groups specified by the initialization parameter ASM\_DISKGROUPS. If you have not entered a value for ASM\_DISKGROUPS, then the ASM instance starts and Oracle displays an error that no disk groups were mounted.

The Cluster Synchronization Services (CSS) daemon provides cluster services for ASM, communication between the ASM and database instances, and other essential services

## Which type of database file is spread across

All types of files are spread across, depend on which type or latency, there are several ways to strip or divide the file.

## Net effect of the command that does not define the REBALANCE option

Oracle will automatically rebalance the diskgroup before drop/add disk to diskgroup.

## ALTER DISKGROUP MOUNT RESTRICTED.

In the RESTRICTED mode, the disk group is mounted in single-instance exclusive mode. No other Oracle ASM instance in the same cluster can mount that disk group. In this mode the disk group is not usable by any Oracle ASM client.

## check\_diskgroup\_clause

The check\_diskgroup\_clause lets you verify the internal consistency of Automatic Storage Management disk group metadata. The disk group must be mounted. Automatic Storage Management displays summary errors and writes the details of the detected errors in the alert log.

* The CHECK keyword performs the following operations:
* Checks the consistency of the disk.
* Cross checks all the file extent maps and allocation tables for consistently.
* Checks that the alias metadata directory and file directory are linked correctly.
* Checks that the alias directory tree is linked correctly.
* Checks that ASM metadata directories do not have unreachable allocated blocks.

## STARTUP MOUNT FORCE | NOFORCE Use these clauses to determine the circumstances under which the disk groups are mounted.

In the FORCE mode, ASM attempts to mount the disk group even if it cannot discover all of the devices that belong to the disk group. This setting is useful if some of the disks in a normal or high redundancy disk group became unavailable while the disk group was dismounted. When MOUNT FORCE succeeds, ASM takes the missing disks offline.

If ASM discovers all of the disks in the disk group, then MOUNT FORCE fails. Therefore, use the MOUNT FORCE setting only if some disks are unavailable. Otherwise, use NOFORCE.

## Metadata Backup and Restore

It records all the diskgroups, disks, directories, the disk attributes and so on. By default, this file records all the diskgroups. If you want to backup only a specific diskgroup, you can use the -g option. In addition, you can use -b option to create a specific named file.

## ASM\_POWER\_LIMIT

ASM\_POWER\_LIMIT specifies the maximum power on an Automatic Storage Management instance for disk rebalancing. The higher the limit, the faster rebalancing will complete. Lower values will take longer, but consume fewer processing and I/O resources.

## Optimal Flexible Architecture File Mapping

| **Directory** | **Description** |
| --- | --- |
| / | Root directory |
| /u01/ | User data mount point 1 |
| /u01/app/ | Subtree for application software |
| /u01/app/oracle/ | Oracle Base directory |
| /u01/app/oracle/admin/ | Subtree for database administration files |
| /u01/app/oracle/admin/TAR | Subtree for support log files |
| /u01/app/oracle/admin/db\_name1/ | admin subtree for db\_name1 database |
| /u01/app/oracle/admin/db\_name2/ | admin subtree for db\_name2 database |
| /u01/app/oracle/doc/ | Online documentation |
| /u01/app/oracle/fast\_recovery\_area/ | Subtree for recovery files |
| /u01/app/oracle/fast\_recovery\_area/db\_name1 | Recovery files for db\_name1 database |
| /u01/app/oracle/fast\_recovery\_area/db\_name2 | Recovery files for db\_name2 database |
| /u02/app/oracle/oradata  /u03/app/oracle/oradata  /u04/app/oracle/oradata | Oracle data directory |
| /u01/app/oracle/product/ | Distribution files |
| /u01/app/oracle/product/11.2.0/dbhome\_1 | Oracle home directory for Oracle Database, for user oracle |
| /u01/app/oracle/product/11.2.0/grid | Oracle home directory for Oracle Grid Infrastructure for a standalone server, for user oracle |
| /u01/app/kjf/ | Oracle base directory for user kjf |
| /u01/app/edm/ | Oracle base directory for user edm |

## Table D-9 Hierarchical File Mapping for Log Files in an Optimal Flexible Architecture Installation

| **Directory** | **Description** |
| --- | --- |
| /u01/app/oracle/admin/TAR | Subtree for support log files |
| /u01/app/oracle/admin/orcl/arch/\* | Archived log files |
| /u01/app/oracle/admin/orcl/create/ | Contains the database creation log files |
| /u01/app/oracle/oradata/orcl/\*.log | Redo log files |
| /u01/app/oracle/admin/orcl/dpdump/ | Contains the data pump file dp.log |
| /u01/app/oracle/diag | Contains all database, listener, sqlnet and other diagnostic logs |
| /u01/app/oracle/audit | Contains all audit logs |
| /u01/app/oracle/cfgtoollogs | Contains logs for configuration assistants such as Oracle Database Configuration Assistant, Database Upgrade Assistant, and Oracle Net Configuration Assistant |

## disk\_offline\_clause

Use the disk\_offline\_clause to take one or more disks offline. This clause fails if the redundancy level of the disk group would be violated by taking the specified disks offline.

By default, Oracle ASM drops a disk shortly after it is taken offline. You can delay this operation by specifying the timeout\_clause, which gives you the opportunity to repair the disk and bring it back online. You can specify the timeout value in units of minute or hour. If you omit the unit, then the default is hour.

## Disk Group Compatibility

### COMPATIBLE.ASM

The value for the disk group COMPATIBLE.ASM attribute determines the minimum software version for an Oracle ASM instance that can use the disk group. *This setting also affects the format of the data structures for the Oracle ASM metadata on the disk*. The format of other file contents is determined by Oracle ASM Dynamic Volume Manager (Oracle ADVM) and the database instance.

### COMPATIBLE.RDBMS

The value for the disk group COMPATIBLE.RDBMS attribute determines the minimum COMPATIBLE database initialization parameter setting for any database instance that is allowed to use the disk group.

### COMPATIBLE.ADVM

The value for the disk group COMPATIBLE.ADVM attribute determines whether the disk group can contain Oracle ASM volumes. The value must be set to 11.2 or higher. Before setting this attribute, the COMPATIBLE.ASM value must be 11.2 or higher. Also, the Oracle ADVM volume drivers must be loaded in the supported environment.

By default, the value of the COMPATIBLE.ADVM attribute is empty until set.

The value of COMPATIBLE.ASM must always be greater than or equal to the value of COMPATIBLE.RDBMS and COMPATIBLE.ADVM.

## Features enabled by disk group compatibility attribute settings

| **Disk Group Features Enabled** | **COMPATIBLE.ASM** | **COMPATIBLE.RDBMS** | **COMPATIBLE.ADVM** |
| --- | --- | --- | --- |
| Support for larger AU sizes (32 or 64 MB) | >= 11.1 | >= 11.1 | n/a |
| Attributes are displayed in the V$ASM\_ATTRIBUTE view | >= 11.1 | n/a | n/a |
| Fast mirror resync | >= 11.1 | >= 11.1 | n/a |
| Variable size extents | >= 11.1 | >= 11.1 | n/a |
| Exadata storage | >= 11.1.0.7 | >= 11.1.0.7 | n/a |
| Intelligent Data Placement | >= 11.2 | >= 11.2 | n/a |
| OCR and voting files in a disk group | >= 11.2 | n/a | n/a |
| Sector size set to nondefault value | >= 11.2 | >= 11.2 | n/a |
| Oracle ASM SPFILE in a disk group | >= 11.2 | n/a | n/a |
| Oracle ASM File Access Control | >= 11.2 | >= 11.2 | n/a |
| Volumes in disk groups | >= 11.2 | n/a | >= 11.2 |
| ASM\_POWER\_LIMIT value up to 1024 | >= 11.2.0.2 | n/a | n/a |
| Encryption, replication, security, tagging on Linux systems | >= 11.2.0.2 | n/a | >= 11.2.0.2 |
| Encryption, replication, security, tagging on Windows systems | >= 11.2.0.3 | n/a | >= 11.2.0.3 |
| Read-write snapshots | >= 11.2.0.3 | n/a | >= 11.2.0.3 |

## The redundancy levels are:

### External redundancy

Oracle ASM does not provide mirroring redundancy and relies on the storage system to provide RAID functionality. Any write error causes a forced dismount of the disk group. All disks must be located to successfully mount the disk group.

### Normal redundancy

Oracle ASM provides two-way mirroring by default, which means that all files are mirrored so that there are two copies of every [extent](http://docs.oracle.com/cd/E11882_01/server.112/e18951/ostmg_gloss.htm#BABFCABI). A loss of one Oracle ASM disk is tolerated. You can optionally choose three-way or unprotected mirroring.

A file specified with HIGH redundancy (three-way mirroring) in a NORMAL redundancy disk group provides additional protection from a bad disk sector, not protection from a disk failure.

### High redundancy

Oracle ASM provides triple mirroring by default. A loss of two Oracle ASM disks in different failure groups is tolerated.

## Allocation Units

When you create a disk group, you can set the Oracle ASM allocation unit size with the AU\_SIZE disk group attribute. *The values can be 1, 2, 4, 8, 16, 32, or 64 MB*, depending on the specific disk group compatibility level. Larger AU sizes typically provide performance advantages for data warehouse applications that use large sequential reads.

*This attribute can only be set when creating a disk group.*

## Options for the md\_restore command

| **Option** | **Description** |
| --- | --- |
| backup\_file | Reads the metadata information from backup\_file. |
| --silent | Ignore errors. Typically, if md\_restore encounters an error, it stops. Specifying this flag ignores any errors. |
| --full | Specifies to create a disk group and restore metadata. |
| --nodg | Specifies to restore metadata only. |
| --newdg -o old\_diskgroup:new\_diskgroup] | Specifies to create a disk group with a different name when restoring metadata. The -o option is required with --newdg. |
| -S sql\_script\_file | Write SQL commands to the specified SQL script file instead of executing the commands. |
| -G diskgroup | Select the disk groups to be restored. If no disk groups are defined, then all disk groups are restored. |

## ASM\_DISKGROUPS

| **Property** | **Description** |
| --- | --- |
| Parameter type | String |
| Syntax | ASM\_DISKGROUPS = diskgroup [, diskgroup ] ... |
| Default value | There is no default value. |
| Modifiable | ALTER SYSTEM |
| Range of values | Comma-separated list of strings, up to 30 characters |
| Oracle RAC | Multiple instances can have different values. |

## ASM\_DISKSTRING

| **Property** | **Description** |
| --- | --- |
| **Parameter type** | String |
| **Syntax** | ASM\_DISKSTRING = discovery\_string [, discovery\_string ] ... |
| **Default value** | The null string; Automatic Storage Management discovery finds all disks in an operating system-specific location to which the Automatic Storage Management instance has read/write access. |
| **Modifiable** | ALTER SYSTEM |
| **Real Application Clusters** | Multiple instances can have different values. Different nodes might see the same disks under different names; however, each instance must be able to use its ASM\_DISKSTRING to discover the same physical media as the other nodes in the cluster. |

## DISKGROUP MOUNT FORCE | NOFORCE

In the FORCE mode

ASM attempts to mount the disk group even if it cannot discover all of the devices that belong to the disk group. This setting is useful if some of the disks in a normal or high redundancy disk group became unavailable while the disk group was dismounted. When MOUNT FORCE succeeds, ASM takes the missing disks offline.

If ASM discovers all of the disks in the disk group, then MOUNT FORCE fails. Therefore, use the MOUNT FORCE setting only if some disks are unavailable. Otherwise, use NOFORCE.

In normal- and high-redundancy disk groups, disks from one failure group can be unavailable and MOUNT FORCE will succeed. Also in high-redundancy disk groups, two disks in two different failure groups can be unavailable and MOUNT FORCE will succeed. Any other combination of unavailable disks causes the operation to fail, because ASM cannot guarantee that a valid copy of all user data or metadata exists on the available disks.

In the NOFORCE mode

ASM does not attempt to mount the disk group unless it can discover all the member disks. This is the default.

## DG\_DROP\_TIME

There is no such thing DG\_DROP\_TIME in diskgroup's attribute list.

## Extents

*Variable size extents enable support for larger ASM datafiles, reduce SGA memory requirements for very large databases, and improve performance for file create and open operations.*

The contents of Oracle ASM files are stored in a disk group as a set, or collection, of [extent](http://docs.oracle.com/cd/E11882_01/server.112/e18951/ostmg_gloss.htm#BABFCABI)s that are stored on individual disks within disk groups. Each extent resides on an individual disk. *Extents consist of one or more allocation units (AU).* To accommodate increasingly larger files, Oracle ASM uses variable size extents.

The initial extent size equals the disk group allocation unit size and it increases by a factor of 4 or 16 at predefined thresholds. *This feature is automatic for newly created and resized data files when specific disk group compatibility attributes are set to 11.1 or higher*. For information about compatibility attributes, see ["Disk Group Compatibility"](http://docs.oracle.com/cd/E11882_01/server.112/e18951/asmdiskgrps.htm#CHDDIGBJ).

*The extent size of a file varies as follows:*

*Extent size always equals the disk group AU size for the first 20000 extent sets (0 - 19999).*

*Extent size equals 4\*AU size for the next 20000 extent sets (20000 - 39999).*

*Extent size equals 16\*AU size for the next 20000 and higher extent sets (40000+).*

## Table 13-2 Database Shutdown Modes

| **Database Behavior** | **ABORT** | **IMMEDIATE** | **TRANSACTIONAL** | **NORMAL** |
| --- | --- | --- | --- | --- |
| Permits new user connections | No | No | No | No |
| Waits until current sessions end | No | No | No | Yes |
| Waits until current transactions end | No | No | Yes | Yes |
| Performs a [checkpoint](http://docs.oracle.com/cd/E25054_01/server.1111/e25789/glossary.htm#CHDDBJEH) and closes open files | No | Yes | Yes | Yes |

## Shutting Down an ASM Instance

### NORMAL Clause

ASM waits for any in-progress SQL to complete before performing an orderly dismount of all of the disk groups and shutting down the ASM instance. Before the instance is shut down, ASM waits for all of the currently connected users to disconnect from the instance. *If any database instances are connected to the ASM instance, then the SHUTDOWN command returns an error and leaves the ASM instance running*. NORMAL is the default shutdown mode.

### IMMEDIATE or TRANSACTIONAL Clause

ASM waits for any in-progress SQL to complete before performing an orderly dismount of all of the disk groups and shutting down the ASM instance. ASM does not wait for users currently connected to the instance to disconnect*. If any database instances are connected to the ASM instance, then the SHUTDOWN command returns an error and leaves the ASM instance running.* Because the ASM instance does not contain any transactions, the TRANSACTIONAL mode is the same as the IMMEDIATE mode.

### ABORT Clause

*The ASM instance immediately shuts down without the orderly dismount of disk groups. This causes recovery to occur upon the next ASM startup. If any database instance is connected to the ASM instance, then the database instance aborts.*

## About ASM Instances

An ASM instance is built on the same technology as an Oracle Database instance. *An ASM instance has a System Global Area (SGA) and background processes that are similar to those of Oracle Database*. However, because ASM performs fewer tasks than a database, an ASM SGA is much smaller than a database SGA. In addition, ASM has a minimal performance effect on a server. *ASM instances mount disk groups to make ASM files available to database instances; ASM instances do not mount databases.*

## LSDSK - Check disk infomation even if ASM instance is down.

|  |  |
| --- | --- |
| -I | Scans disk headers for information rather than extracting the information from an ASM instance. *This option forces the non-connected mode.* |
| -d | Restricts results to only those disks that belong to the group specified by disk\_group\_name. |

## State of the disk group relative to the instance:

|  |  |
| --- | --- |
|  | * CONNECTED - Disk group is in use by the database instance * BROKEN - Database instance lost connectivity to the Automatic Storage Management instance that mounted the disk group * UNKNOWN - Automatic Storage Management instance has never attempted to mount the disk group * DISMOUNTED - Disk group was cleanly dismounted by the Automatic Storage Management instance following a successful mount * MOUNTED - Instance is successfully serving the disk group to its database clients |

## remap

### Purpose

Marks a range of blocks as unusable on the disk and relocates any data allocated in that range.

The remap command only relocates blocks. It does not correct or repair blocks that contain corrupted contents. The command uses a physical block size based on the SECTOR\_SIZE disk group attribute.

## ASMB process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASMB | ASM Background Process | Communicates with the ASM instance, managing storage and providing statistics | ASMB runs in ASM instances when the ASMCMD cp command runs or when the database instance first starts if the server parameter file is stored in ASM. ASMB also runs with Oracle Cluster Registry on ASM. | Database and ASM instances |

## **Automatic Memory Management for Oracle ASM**

Automatic memory management automatically manages the memory-related parameters for both Oracle ASM and database instances with the MEMORY\_TARGET parameter. Automatic memory management is enabled by default on an Oracle ASM instance, even when the MEMORY\_TARGET parameter is not explicitly set.

## ASM Preferred Mirror Read

Mirroring is done to ensure the protection from data loss. ASM failure groups in Oracle Database 10g always reads the primary copy of a mirrored ASM data extent. If a *node is allowed to read the data from local or from closest node inspite that data is secondary, then it can improve the performance by reducing network traffic*. Oracle Database 11g can do this by configuring preferred mirror read using the new ASM\_PREFERRED\_READ\_FAILURE\_GROUPS initialization parameter to specify a list of preferred mirror read names.

## UTLRP.SQL

The utlrp.sql script simply calls the utlprp.sql script with a command line parameter of "0". The utlprp.sql accepts a single integer parameter that indicates the level of parallelism as follows.

0 - The level of parallelism is derived based on the CPU\_COUNT parameter.

1 - The recompilation is run serially, one object at a time.

N - The recompilation is run in parallel with "N" number of threads.

## DISK\_REPAIR\_TIME

*Altering the DISK\_REPAIR\_TIME attribute has no effect on offline disks*. The new value is used for any disks that go offline after the attribute is updated. You can confirm this behavior by viewing the Oracle ASM alert log.

# Topic 2: Configuring for Recoverability

## Recover command RMAN

|  |  |
| --- | --- |
| [RECOVER](http://docs.oracle.com/cd/E11882_01/backup.112/e10643/rcmsynta2001.htm#i84135) | Apply redo log files and incremental backups to data files or data blocks restored from backup or dataflow copies, to update them to a specified time. |

All redo or incremental changes required for the recovery must exist on disk or in SBT. If RMAN needs to restore incremental backups or archived redo log files during recovery, then you must either have automatic channels configured or manually allocate channels of the same type that created these backups.

## incarnation

A separate version of a database. The incarnation of the database changes when you open it with the RESETLOGS option, but you can recover backups from a prior incarnation so long as the necessary redo is available.

Connect RMAN to the target database and perform a flashback to the SCN immediately before the RESETLOGS.

Use the following form of the FLASHBACK DATABASE command:

FLASHBACK DATABASE TO BEFORE RESETLOGS;

## About Block Change Tracking

If block change tracking is enabled on a primary or standby database, then RMAN uses a [block change tracking file](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#CHDBFBFB) to identify changed blocks for incremental backups. By reading this small bitmap file to determine which blocks changed, RMAN avoids having to scan every block in the data file that it is backing up.

## expired backup

A backup whose status in the RMAN repository is EXPIRED, which means that the backup was not found. RMAN marks backups and copies as expired when you run a CROSSCHECK command and the files are absent or inaccessible.

## DB\_BLOCK\_CHECKSUM

DB\_BLOCK\_CHECKSUM determines whether DBWn and the direct loader will calculate a checksum (a number calculated from all the bytes stored in the block) and store it in the cache header of every data block when writing it to disk. Checksums are verified when a block is read - only if this parameter is TYPICAL or FULL and the last write of the block stored a checksum. In FULL mode, Oracle also verifies the checksum before a change application from update/delete statements and recomputes it after the change is applied. In addition, Oracle gives every log block a checksum before writing it to the current log.

*If this parameter is set to OFF, DBWn calculates checksums only for the SYSTEM tablespace, but not for user tablespaces.*

Checksums allow Oracle to detect corruption caused by underlying disks, storage systems, or I/O systems. If set to FULL, DB\_BLOCK\_CHECKSUM also catches in-memory corruptions and stops them from making it to the disk. Turning on this feature in TYPICAL mode causes only an additional 1% to 2% overhead. In the FULL mode it causes 4% to 5% overhead. Oracle recommends that you set DB\_BLOCK\_CHECKSUM to TYPICAL. *For backward compatibility we preserve use of TRUE (implying TYPICAL) and FALSE (implying OFF) values.*

## Multilevel Incremental Backups

RMAN can create multilevel incremental backups. Each incremental level is denoted by a value of 0 or 1. A [level 0 incremental backup](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#CHDCBDHA), which is the base for subsequent incremental backups, copies all blocks containing data. You can create a level 0 database backup as backup sets or image copies.

The only difference between a level 0 incremental backup and a full backup is that a full backup is never included in an incremental strategy. Thus, an incremental level 0 backup is a full backup that happens to be the parent of incremental backups whose level is greater than 0.

A level 1 incremental backup can be either of the following types:

A [differential incremental backup](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i432588), which backs up all blocks changed after the most recent incremental backup at level 1 or 0

A [cumulative incremental backup](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i432455), which backs up all blocks changed after the most recent incremental backup at level 0

## [flash recovery area](http://docs.oracle.com/cd/B28359_01/backup.111/b28270/glossary.htm#i999320)

An important part of RMAN maintenance is deleting backups that are no longer needed*. If you configure a* [*flash recovery area*](http://docs.oracle.com/cd/B28359_01/backup.111/b28270/glossary.htm#i999320)*, then the database automatically deletes unneeded files in this area automatically*; even so, you may want to delete backups and copies from tape. You may even need to delete an entire database. You can use an RMAN command to perform these tasks.

## TEMP FILES CORRUPT

Allow database to continue running, add new tempfile to TEMP TBSP with a new name, then delete the corrupted tempfile.

## BACKUP STATUS

A backup is obsolete when REPORT OBSOLETE or DELETE OBSOLETE determines, based on the user-defined retention policy, that it is not needed for recovery. A backup is considered an [expired backup](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#CHDGICAB) only when RMAN performs a crosscheck and cannot find the file. In short, obsolete means a file is not needed, whereas expired means it is not found.

## Validating Database Files with BACKUP VALIDATE

You can use the BACKUP VALIDATE command to do the following:

* Check datafiles for physical and logical block corruption
* Confirm that all database files exist and are in the correct locations
* You can use the [VALIDATE](http://docs.oracle.com/cd/E11882_01/backup.112/e10643/rcmsynta2024.htm#RCMRF162) command to manually check for physical and logical corruptions in database files. This command performs the same types of checks as BACKUP VALIDATE, but VALIDATE can check a larger selection of objects. For example, you can validate individual blocks with the VALIDATE DATAFILE ... BLOCK command.
* When validating whole files, RMAN checks every block of the input files. If the backup validation discovers previously unmarked corrupt blocks, then RMAN updates the V$DATABASE\_BLOCK\_CORRUPTION view with rows describing the corruptions.
* The following command repairs all physically corrupted blocks recorded in the view: RMAN> RECOVER CORRUPTION LIST;

## LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST

LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST defines the minimum number of destinations that must succeed in order for the online logfile to be available for reuse.

* If you are using the LOG\_ARCHIVE\_DEST\_n parameters and automatic archiving is enabled, then the value of this parameter cannot exceed the number of enabled, valid destinations specified as MANDATORY plus the number of enabled, valid destinations that are configured with the OPTIONAL and LOCATION attributes.
* If you are using LOG\_ARCHIVE\_DEST and LOG\_ARCHIVE\_DUPLEX\_DEST and automatic archiving is enabled, a value of 1 specifies that the destination specified in LOG\_ARCHIVE\_DEST must succeed. A value of 2 specifies that the destinations specified in both parameters must succeed.

If the value of this parameter is less than the number of enabled, valid MANDATORY destinations, this parameter is ignored in favor of the MANDATORY destination count.

## INTRABLOCK VS INTERBLOCK

## RECOVER

Purpose

* Use the RECOVER command to perform one of the following distinct tasks:
* Perform complete recovery of the whole database or one or more restored data files
* Perform point-in-time recovery of a database (DBPITR) or tablespace (TSPITR)
* Apply incremental backups to a data file image copy (not a restored data file) to roll it forward in time
* Recover a corrupt data block or set of data blocks within a data file

|  |  |
| --- | --- |
| [fast recovery area](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i999320) | A disk location that you can use to store recovery-related files such as control file and online redo log copies, archived redo logs, [flashback logs](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#CHDFHAAE), and RMAN backups. Oracle Database and RMAN manage the files in the fast recovery area automatically. |

## Losing an Active Online Redo Log Group

If the database is still running and the lost active redo log is not the current log, then issue the ALTER SYSTEM CHECKPOINT statement. If the operation is successful, then the active redo log becomes inactive, and you can follow the procedure in ["Losing an Inactive Online Redo Log Group"](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/osadvsce.htm#CACEDJAE). If the operation is unsuccessful, or if your database has halted, then perform one of procedures in this section, depending on the archiving mode.

## Clearing Inactive, Archived Redo

You can clear an inactive redo log group when the database is open or closed. The procedure depends on whether the damaged group has been archived.

## Clearing Inactive, Unarchived Redo

Clearing a not-yet-archived redo log allows it to be reused without archiving it. This action makes backups unusable if they were started before the last change in the log, unless the file was taken offline before the first change in the log. Hence, if you need the cleared log file for recovery of a backup, then you cannot recover that backup. Clearing a not-yet-archived-redo-log, prevents complete recovery from backups due to the missing log.

# Topic 3: Oracle RMAN Catalog

## RMAN Scripts

Execute CREATE SCRIPT only at the RMAN prompt. RMAN must be connected to a target database and a recovery catalog. The recovery catalog database must be open.

If GLOBAL is specified, then a global script with this name must not exist in the recovery catalog. If GLOBAL is not specified, then a local script must not exist with the same name for the same target database. In you do not meet these prerequisites, then RMAN returns error RMAN-20401.

A stored script may be local or global. A local script is created for the current target database only, whereas a global script is available for use with any database registered in the recovery catalog.

It is permissible to create a global script with the same name as a local script, or a local script with the same name as a global script.

## Overview of the Recovery Catalog

### Purpose of the Recovery Catalog

A [recovery catalog](http://docs.oracle.com/cd/E25054_01/backup.1111/e10642/glossary.htm#i433280) is a database schema used by RMAN to store metadata about one or more Oracle databases. Typically, you store the catalog in a dedicated database. A recovery catalog provides the following benefits:

* A recovery catalog creates redundancy for the RMAN repository stored in the control file of each [target database](http://docs.oracle.com/cd/E25054_01/backup.1111/e10642/glossary.htm#i433583). The recovery catalog serves as a secondary metadata repository. If the target control file and all backups are lost, then the RMAN metadata still exists in the recovery catalog.
* A recovery catalog centralizes metadata for all your target databases. Storing the metadata in a single place makes reporting and administration tasks easier to perform.
* A recovery catalog can store metadata history much longer than the control file. This capability is useful if you must do a recovery that goes further back in time than the history in the control file. The added complexity of managing a recovery catalog database can be offset by the convenience of having the extended backup history available.

Some RMAN features function only when you use a recovery catalog. For example, you can store RMAN scripts in a recovery catalog. The chief advantage of a [stored script](http://docs.oracle.com/cd/E25054_01/backup.1111/e10642/glossary.htm#i433542) is that it is available to any RMAN client that can connect to the target database and recovery catalog. Command files are only available if the RMAN client has access to the file system on which they are stored.

A recovery catalog is required when you use RMAN in a Data Guard environment. By storing backup metadata for all primary and standby databases, the catalog enables you to offload backup tasks to one standby database while enabling you to restore backups on other databases in the environment.

### Basic Concepts for the Recovery Catalog

The recovery catalog contains metadata about RMAN operations for each registered target database. When RMAN is connected to a recovery catalog, RMAN obtains its metadata exclusively from the catalog. The catalog includes the following types of metadata:

* Datafile and archived redo log backup sets and backup pieces
* Datafile copies
* Archived redo logs and their copies
* Database structure (tablespaces and datafiles)
* Stored scripts, which are named user-created sequences of RMAN commands
* Persistent RMAN configuration settings

## IMPORT CATALOG

If the operation fails in the middle of the import, then the import is rolled back. Thus, a partial import is not permitted. The unregister operation is separate from the import. *By default, the imported database IDs are unregistered from the source recovery catalog schema after a successful import.*

Stored scripts are either global or local. It is possible for global scripts, but not local scripts, to have name conflicts during import because the destination schema already contains the script name. In this case, *RMAN renames the global script name to COPY OF script\_name. For example, RMAN renames bp\_cmd to COPY OF bp\_cmd.*

If the renamed global script is still not unique, then RMAN renames it to COPY(2) OF script\_name. If this script name also exists, then RMAN renames the script to COPY(3) OF script\_name. RMAN continues the COPY(n) OF pattern until the script is uniquely named.

## Centralization of Metadata in a Base Recovery Catalog

The owner of a centralized recovery catalog, which is also called the [base recovery catalog](http://docs.oracle.com/cd/E25054_01/backup.1111/e10642/glossary.htm#CHDGBFBF), can grant or revoke restricted access to the catalog to other database users. Each restricted user has full read/write access to his own metadata, which is called a [virtual private catalog](http://docs.oracle.com/cd/E25054_01/backup.1111/e10642/glossary.htm#CHDBECHC). The RMAN metadata is stored in the schema of the virtual private catalog owner. The owner of the base recovery catalog determines which objects each virtual private catalog user can access.

# Topic 4: Configuring Backup

## Using the RATE Parameter to Control Disk Bandwidth Usage

The RATE parameter specifies the bytes/second that RMAN reads on this channel. Use this parameter to set an upper limit for bytes read so that RMAN does not consume excessive disk bandwidth and degrade online performance.

For example, set RATE=1500K. If each disk drive delivers 3 MB/second, then RMAN leaves some disk bandwidth available to the online system.

## Configuring a Recovery Window-Based Retention Policy

The RECOVERY WINDOW parameter of the CONFIGURE command specifies the number of days between the current time and the earliest point of recoverability. RMAN does not consider any full or [level 0 incremental backup](http://docs.oracle.com/cd/B28359_01/backup.111/b28270/glossary.htm#CHDCBDHA) as obsolete if it falls within the recovery window. Additionally, RMAN retains all archived logs and level 1 incremental backups that are needed to recover to a random point within the window.

### [Transparent Encryption of Backups](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmconfa.htm#CHDFFFEH)

This is the default mode and uses the Oracle wallet. A wallet is a password-protected container used to store authentication and signing credentials, including private keys, certificates, and trusted certificates needed by SSL.

### [Password Encryption of Backups](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmconfa.htm#CHDFADGB)

This mode uses only password protection. You must provide a password when creating and restoring encrypted backups.

### [Dual Mode Encryption of Backups](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmconfa.htm#CHDGFGJD)

This mode requires either the wallet or a password.

# Topic 5: Using RMAN to Create Backup

## REPORT NEED BACKUP

Reports which database files need to be backed up to meet a configured or specified retention policy

## Migrating a Database into ASM - Migrate THINGS to ASM

http://docs.oracle.com/cd/B19306\_01/backup.102/b14191/rcmasm.htm#i1022780

## Purpose of RMAN Reporting

As part of your backup and recovery strategy, you should periodically run reports that indicate what you have backed up. You should also determine which datafiles need backups or which files have not been backed up recently. Also, you can preview which backups RMAN would need to restore if a problem were to occur.

Another important aspect of backup and recovery is monitoring space usage. If you back up to disk, then it is possible for the disk to fill, which can create performance problems or even cause the database to halt. You can use RMAN to determine whether a backups is an [obsolete backup](http://docs.oracle.com/cd/B28359_01/backup.111/b28270/glossary.htm#CHDIAGEJ) and can therefore be deleted.

You may also need to obtain historical information about RMAN jobs. For example, you may want to know how many backup jobs have been issued, the status of each backup job (for example, whether it failed or completed), when a job started and finished, and what type of backup was performed.

## RMAN in Debug Mode

If you understand how to correctly read the error stack of the RMAN, the information given at any error is enough to troubleshoot the underlying failed operation and command. However, at times it is required to dig much deeper than what is available on the outer side.  For example, you want to check how the I/O calls for a backup and how restore and recover operations are working. For this, the standard error messaging or logging will not be helpful.

Similarly, if a channel is dying abruptly when allocated, reasoning for that will not be clear from the standard logging. Oracle database has capabilities to enable tracing and debugging sessions for almost all parts of it. In the same way, RMAN can also be run in the debug mode where the flow of the underlying and somewhat hidden operations can be revealed.

The debug optionis a resource-consuming option, both in the terms of physical and computational resources. Even a simple command like report, list would be very slow when run with the debug option. The resultant trace files also come up with a huge size and the contents also do not make much sense if you are not very familiar with reading raw trace files like trace files of 10046 and 10053 trace events!  For these reasons, an explanation of the contents of the trace file are beyond the scope of this book, but an introduction on how to enable the debug option and its various options will be given.

## RMAN BACKUP

When you execute the BACKUP command in RMAN, the output is always either one or more backup sets or one or more image copies. A [backup set](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i432144) is an RMAN-specific proprietary format, whereas an [image copy](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i432704) is a bit-for-bit copy of a file. By default, RMAN creates backup sets.

## Multiple Copies of RMAN Backups

In RMAN, you can make multiple, identical copies of backups in the following ways:

* Duplex backups with the BACKUP ... COPIES command, in which case RMAN creates multiple copies of each backup set
* Back up your files as backup sets or image copies, and then back up the backup sets or image copies with the RMAN BACKUP BACKUPSET or BACKUP COPY OF commands
* REPORT NEED BACKUP RECOVERY WINDOW OF n DAYS

Displays objects requiring backup to satisfy a recovery window-based retention policy

* REPORT NEED BACKUP REDUNDANCY n

Displays objects requiring backup to satisfy a redundancy-based retention policy

* REPORT NEED BACKUP DAYS n

Displays files that require more than n days' worth of archived redo log files for recovery

* REPORT NEED BACKUP INCREMENTAL n

Displays files that require application of more than n incremental backups for recovery

## Basic Concepts of Archival BSackups

You can exempt a backup from the retention policy by using the KEEP option with the BACKUP command. You can also use the KEEP and NOKEEP options of the [CHANGE](http://docs.oracle.com/cd/E11882_01/backup.112/e10643/rcmsynta009.htm#RCMRF110) command to change the status of an existing backup. Backups with KEEP attributes are valid backups that can be recovered like any other backups.

You can specify an end date for an archival backup with the [KEEP UNTIL TIME](http://docs.oracle.com/cd/E11882_01/backup.112/e10643/rcmsubcl019.htm#RCMRF160) clause, or specify that the backup should be kept FOREVER. If you specify UNTIL, then RMAN marks the backup as obsolete when the UNTIL time has passed, regardless of any configured retention policy. For example, if you specify KEEP UNTIL TIME '01-JAN-08', then the backup is obsolete one second after midnight on January 1. If you specify an UNTIL TIME of 9:00 p.m, then the backup is obsolete at 9:01 p.m.

When you specify KEEP on the BACKUP command, RMAN generates multiple backup sets. Note the following characteristics of the BACKUP ... KEEP command:

* It automatically backs up the data files, control file (even if the [control file autobackup](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i432379) is disabled), and the server parameter file.
* It automatically generates an archived redo log backup to ensure that the database backup can be recovered to a consistent state.
* If the FORMAT, POOL, or TAG parameters are specified, then they are used for all backups. For this reason, the FORMAT string must allow for the creation of multiple backup pieces. Specifying the %U substitution variable is the easiest way to meet this requirement.
* It supports an optional RESTORE POINT clause that creates a [normal restore point](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#CHDIEBAH), which is a label for an SCN to which the backup must be recovered to be made consistent. The SCN is captured just after the data file backups complete. RMAN resynchronizes restore points with the [recovery catalog](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i433280) and maintains the restore points as long as the backup exists. ["Listing Restore Points"](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/flashdb.htm#CFHGBFEI) explains how to display restore points.

## Controlling RMAN Behavior when Backup Window Ends with PARTIAL

By default, when a BACKUP... DURATION command runs out of time before the backup completes, RMAN reports an error. (The effect of this is that if the command is running in a RUN block, the RUN block terminates.) You can control this behavior by adding the PARTIAL option to the BACKUP... DURATION command, as in this example:

BACKUP DURATION 4:00 PARTIAL TABLESPACE users FILESPERSET 1;

When PARTIAL is used, no error is reported when a backup command is interrupted due to the end of the backup window. Instead, a message showing which files could not be backed will be displayed. If the BACKUP command is part of a RUN block, then the remaining commands in the RUN block will continue to execute.

When using DURATION the least recently backed up files are backed up first. Thus, if you retry a job that was interrupted when the available duration expired, each successive attempt covers more of the files needing backup.

Note also the use of FILESPERSET 1 in this example. With this option, RMAN puts each file into its own backupset. This way, when a backup is interrupted at the end of the backup window, only the backup of the file currently being backed up is lost. All backup sets completed during the window are saved, minimizing the lost work due to the end of the backup window.

## RMAN multiplexing

RMAN multiplexing is determined by several factors. For example, the FILESPERSET parameter of the BACKUP command determines how many datafiles to put in each backup set. The MAXOPENFILES parameter of ALLOCATE CHANNEL or CONFIGURE CHANNEL defines how many datafiles RMAN can read from simultaneously. The basic multiplexing algorithm is as follows:

* Number of files in each backup set

*This number is the minimum of the FILESPERSET setting and the number of files read by each channel. The FILESPERSET default is 64.*

* The [level of multiplexing](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#CHDFBDDG)

*This is the number of input files simultaneously read and then written into the same backup piece. The level of multiplexing is the minimum of MAXOPENFILES and the number of files in each backup set. The MAXOPENFILES default is 8.*

# Topic 6: Performing User-Managed Backup and Recovery

## Recovering After the Loss of All Members of an Online Redo Log Group

| **If the group is . . .** | | **Then . . .** | **And you should . . .** |
| --- | --- | --- | --- |
| **Inactive** | It is not needed for crash recovery | | Clear the archived or unarchived group. |
| **Active** | It is needed for crash recovery | | Attempt to issue a checkpoint and clear the log; if impossible, then you must restore a backup and perform incomplete recovery up to the most recent available redo log. |
| **Current** | It is the log that the database is currently writing to | | Attempt to clear the log; if impossible, then you must restore a backup and perform incomplete recovery up to the most recent available redo log. |

# Topic 7: Using RMAN for Recovery

## To recover specific data blocks:

1. Obtain the datafile numbers and block numbers of the corrupted blocks.

The easiest way to locate trace files and the alert log is to connect SQL\*Plus to the target database and execute the following query:

SELECT NAME, VALUE

FROM V$DIAG\_INFO;

1. Start RMAN and connect to the target database, which must be mounted or open.
2. Run the SHOW ALL command to confirm that the appropriate channels are preconfigured.
3. Run the RECOVER ... BLOCK command at the RMAN prompt, specifying the file and block numbers for the corrupted blocks.

The following example recovers two blocks.

RECOVER

DATAFILE 8 BLOCK 13

DATAFILE 2 BLOCK 19;

You can also specify various options to control RMAN behavior. The following example indicates that only backups with the tag mondayam will be used when searching for blocks. You could use the FROM BACKUPSET option to restrict the type of backup that RMAN searches, or the EXCLUDE FLASHBACK LOG option to restrict RMAN from searching the flashback logs.

RECOVER

DATAFILE 8 BLOCK 13

DATAFILE 2 BLOCK 199

FROM TAG mondayam;

## Recovering All Blocks in V$DATABASE\_BLOCK\_CORRUPTION

In this scenario, RMAN automatically recovers all blocks listed in the V$DATABASE\_BLOCK\_CORRUPTION view.

To recover all blocks logged in V$DATABASE\_BLOCK\_CORRUPTION:

1. Start SQL\*Plus and connect to the target database.
2. Query V$DATABASE\_BLOCK\_CORRUPTION to determine whether corrupt blocks exist. For example, execute the following statement:
3. SQL> SELECT \* FROM V$DATABASE\_BLOCK\_CORRUPTION;
4. Start RMAN and connect to the target database.
5. Recover all blocks marked corrupt in V$DATABASE\_BLOCK\_CORRUPTION.

The following command repairs all physically corrupted blocks recorded in the view:

RMAN> RECOVER CORRUPTION LIST;

After the blocks are recovered, the database removes them from V$DATABASE\_BLOCK\_CORRUPTION.

## REPAIR FAILURE

Use the REPAIR FAILURE command to repair database failures identified by the Data Recovery Advisor.

The recommended workflow is to run [LIST](http://docs.oracle.com/cd/B28359_01/backup.111/b28273/rcmsynta027.htm#i82460) FAILURE to display failures, [ADVISE FAILURE](http://docs.oracle.com/cd/B28359_01/backup.111/b28273/rcmsynta003.htm#CHDHBIAB) to display repair options, and REPAIR FAILURE to fix the failures.

## THATS WHY U CAN NOT USE CATALOG IN REPLICATE DB

If you use the procedure in this section, then the DBID for the restored database equals the DBID for the original database. Do not register a test database created in this way in the same recovery catalog as the source database. Because the DBID of the two databases is the same, the metadata for the test database can interfere with RMAN's ability to restore and recover the source database.

# TOPIC 11: FLASHBACK TECHNOLOGY

## Configuring Your Database for Oracle Flashback Transaction Query

To configure your database for the Oracle Flashback Transaction Query feature, you or your database administrator must:

* Ensure that Oracle Database is running with version 10.0 compatibility.
* Enable supplemental logging:

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

## Configuring Your Database for Flashback Transaction

To configure your database for the Flashback Transaction feature, you or your database administrator must:

* With the database mounted but not open, enable ARCHIVELOG:

ALTER DATABASE ARCHIVELOG;

* Open at least one archive log:

ALTER SYSTEM ARCHIVE LOG CURRENT;

* If not done, enable minimal and primary key supplemental logging:

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;

* If you want to track foreign key dependencies, enable foreign key supplemental logging:

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;

## What is recycle bin

The recycle bin is actually a data dictionary table containing information about dropped objects. Dropped tables and any associated objects such as indexes, constraints, nested tables, and the likes are not removed and still occupy space. They continue to count against user space quotas, until specifically purged from the recycle bin or the unlikely situation where they must be purged by the database because of tablespace space constraints.

Each user can be thought of as having his own recycle bin, since unless a user has the SYSDBA privilege, the only objects that the user has access to in the recycle bin are those that the user owns.

## Using Oracle Flashback Version Query

Use Oracle Flashback Version Query to retrieve the different versions of specific rows that existed during a given time interval. A row version is created whenever a COMMIT statement is executed.

Specify Oracle Flashback Version Query using the VERSIONS BETWEEN clause of the SELECT statement. The syntax is:

VERSIONS {BETWEEN {SCN | TIMESTAMP} start AND end}

where start and end are expressions representing the start and end, respectively, of the time interval to be queried. The time interval includes (start and end).

Oracle Flashback Version Query returns a table with a row for each version of the row that existed at any time during the specified time interval. Each row in the table includes pseudocolumns of metadata about the row version, described in [Table 12-1](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#g1020732). This information can reveal when and how a particular change (perhaps erroneous) occurred to your database.

## What shit does Flashback use to "recovery"

They almost use Undo data - Undo Automatic Manage for recovery purpose but FLASHBACK DROP TABLE or FLASHBACK DATABASE orderly use Recycle Bin and flashback logs at FRA for their purpose.

## About the Undo Retention Period

After a transaction is committed, undo data is no longer needed for rollback or transaction recovery purposes. However, for consistent read purposes, long-running queries may require this old undo information for producing older images of data blocks. Furthermore, the success of several Oracle Flashback features can also depend upon the availability of older undo information. For these reasons, it is desirable to retain the old undo information for as long as possible.

When automatic undo management is enabled, there is always a current undo retention period, which is the minimum amount of time that Oracle Database attempts to retain old undo information before overwriting it. Old (committed) undo information that is older than the current undo retention period is said to be expired and its space is available to be overwritten by new transactions. Old undo information with an age that is less than the current undo retention period is said to be unexpired and is retained for consistent read and Oracle Flashback operations.

Oracle Database automatically tunes the undo retention period based on undo tablespace size and system activity. You can optionally specify a minimum undo retention period (in seconds) by setting the UNDO\_RETENTION initialization parameter. The exact impact this parameter on undo retention is as follows:

* *The UNDO\_RETENTION parameter is ignored for a fixed size undo tablespace. The database always tunes the undo retention period for the best possible retention, based on system activity and undo tablespace size. See* [*"Automatic Tuning of Undo Retention"*](http://docs.oracle.com/cd/E18283_01/server.112/e17120/undo002.htm#BABCHJGA) *for more information.*
* For an undo tablespace with the AUTOEXTEND option enabled*, the database attempts to honor the minimum retention period specified by UNDO\_RETENTION. When space is low, instead of overwriting unexpired undo information, the tablespace auto-extends. If the MAXSIZE clause is specified for an auto-extending undo tablespace, when the maximum size is reached, the database may begin to overwrite unexpired undo informatio*n. The UNDOTBS1 tablespace that is automatically created by DBCA is auto-extending.

## Type of flashback

* [Using Oracle Flashback Query (SELECT AS OF)](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#i1008579) for
* [Using Oracle Flashback Version Query](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#i1019938)
* [Using Oracle Flashback Transaction Query](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#i1007455)
* [Using Oracle Flashback Transaction Query with Oracle Flashback Version Query](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#BJFBFGCC)
* [Using DBMS\_FLASHBACK Package](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#i1017341)
* [Using Flashback Transaction](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#BJFIHEIA)
* [Using Flashback Data Archive (Oracle Total Recall)](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#BJFFDCEH)

## Using Flashback Data Archive (Oracle Total Recall)

A Flashback Data Archive provides the ability to track and store transactional changes to a table over its lifetime. A Flashback Data Archive is useful for compliance with record stage policies and audit reports.

A Flashback Data Archive consists of one or more tablespaces or parts thereof. You can have multiple Flashback Data Archives. If you are logged on as SYSDBA, you can specify a default Flashback Data Archive for the system. A Flashback Data Archive is configured with retention time. Data archived in the Flashback Data Archive is retained for the retention time.

## Flashback Data Archive Requirements

1. Flashback Data Archive tablespaces must be managed with automatic segment space management (ASSM).

2. Automatic Undo Management must be enabled.

## Schema Evolution - Flashback Data Archive

Flashback Data Archive guarantees historical data capture for all transactional changes to tracked tables. *In Oracle Database 11g Release 1, the Add Column DDL operation is supported with Flashback Data Archive*. With Oracle Database 11g Release 2, the following DDL operations are supported, with full support for Flashback queries across all associated changes:

• Add, Drop, Rename, Modify Column

• Drop, Truncate Partition

• Rename, Truncate, Drop Table

## Requirements for Enabling Flashback Database

The requirements for enabling Flashback Database are:

* Your database must be running in ARCHIVELOG mode, because archived logs are used in the Flashback Database operation.
* You must have a flash recovery area enabled, because flashback logs can only be stored in the flash recovery area.
* For Real Application Clusters databases, the flash recovery area must be stored in a clustered file system or in ASM.

## Enabling Logging for Flashback Database

1.Start SQL\*Plus and ensure that the database is mounted, but not open. For example:

SQL> SHUTDOWN IMMEDIATE;

SQL> STARTUP MOUNT;

2.Optionally, set theDB\_FLASHBACK\_RETENTION\_TARGET to the length of the desired flashback window in minutes:

SQL> ALTER SYSTEM SET DB\_FLASHBACK\_RETENTION\_TARGET=4320; # 3 days

By default DB\_FLASHBACK\_RETENTION\_TARGET is set to one day (1440 minutes).

3.Enable the Flashback Database feature for the whole database:

SQL> ALTER DATABASE FLASHBACK ON;

## Prerequisites for Flashback Table

To use the Flashback Table feature on one or more tables, use the FLASHBACK TABLE SQL statement with a target time or SCN.

You must have the following privileges to use the Flashback Table feature:

* You must have been granted the FLASHBACK ANY TABLE system privilege or you must have the FLASHBACK object privilege on the table.
* You must have SELECT, INSERT, DELETE, and ALTER privileges on the table.
* To flash back a table to a restore point, you must have the SELECT ANY DICTIONARY or FLASHBACK ANY TABLE system privilege or the SELECT\_CATALOG\_ROLE role.

For an object to be eligible to be flashed back, the following prerequisites must be met:

* The object must not be included the following categories: tables that are part of a cluster, materialized views, Advanced Queuing (AQ) tables, static data dictionary tables, system tables, remote tables, object tables, nested tables, or individual table partitions or subpartitions.
* *The structure of the table must not have been changed between the current time and the target flash back time.*
* *Row movement must be enabled on the table, which indicates that rowids will change after the flashback occurs*.
* *The undo data in the* [*undo tablespace*](http://docs.oracle.com/cd/E18283_01/backup.112/e10642/glossary.htm#i433663) *must extend far enough back in time to satisfy the flashback target time or SCN.*

To ensure that the undo information is retained for Flashback Table operations, Oracle suggests setting the UNDO\_RETENTION parameter to 86400 seconds (24 hours) or greater for the undo tablespace.

## Flashback Data Archive

*You cannot drop the default flashback data archive* unless it has been disabled for all tables for which historical tracking is enabled with this flashback data archive.

### Purpose

Use the DROP FLASHBACK ARCHIVE clause to remove a flashback data archive from the system. This statement removes the flashback data archive and all the historical data in it, but does not drop the tablespaces that were used by the flashback data archive.

## FLASHBACK TABLE ENABLE | DISABLE TRIGGERS

By default, Oracle Database disables all enabled triggers defined on table during the Flashback Table operation and then reenables them after the Flashback Table operation is complete. Specify ENABLE TRIGGERS if you want to override this default behavior and keep the triggers enabled during the Flashback process.

This clause affects only those database triggers defined on table that are already enabled. To enable currently disabled triggers selectively, use the ALTER TABLE ... enable\_disable\_clause before you issue the FLASHBACK TABLE statement with the ENABLE TRIGGERS clause.

## Creating a Flashback Data Archive

Create a Flashback Data Archive with the CREATE FLASHBACK ARCHIVE statement, specifying:

* Name of the Flashback Data Archive
* Name of the first tablespace of the Flashback Data Archive
* (Optional) Maximum amount of space that the Flashback Data Archive can use in the first tablespace
* Retention time (number of days that Flashback Data Archive data for the table is guaranteed to be stored)
* As of Oracle Database 11g Release 2 (11.2.0.4): (Optional) Whether to optimize the storage of data in the history tables maintained in the Flashback Data Archive, using [NO] OPTIMIZE DATA.

The default is NO OPTIMIZE DATA.

# Topic 13: Diag database

## Purpose of Block Media Recovery

You can use [block media recovery](http://docs.oracle.com/cd/E29597_01/backup.1111/e10642/glossary.htm#i432160) to recover one or more corrupt data blocks within a data file. Block media recovery provides the following advantages over [data file media recovery](http://docs.oracle.com/cd/E29597_01/backup.1111/e10642/glossary.htm#i432577):

* Lowers the [mean time to recover (MTTR)](http://docs.oracle.com/cd/E29597_01/backup.1111/e10642/glossary.htm#i432909) because only blocks needing recovery are restored and recovered
* Enables affected data files to remain online during recovery

## About Health Monitor Checks

Health Monitor checks (also known as checkers, health checks, or checks) examine various layers and components of the database. Health checks detect file corruptions, physical and logical block corruptions, undo and redo corruptions, data dictionary corruptions, and more. The health checks generate reports of their findings and, in many cases, recommendations for resolving problems.

## Prerequisites for Block Media Recovery

The following prerequisites apply to the RECOVER ... BLOCK command:

* The target database must run in ARCHIVELOG mode and be open or mounted with a current control file.
* If the target database is a standby database, then it must be in a consistent state, recovery cannot be in session, and the backup must be older than the corrupted file.
* The backups of the data files containing the corrupt blocks must be full or level 0 backups and not proxy copies.
* RMAN can use only archived redo logs for the recovery.
* Flashback Database must be enabled on the target database for RMAN to search the flashback logs for good copies of corrupt blocks.
* The target database must be associated with a real-time query physical standby database for RMAN to search the database for good copies of corrupt blocks.

## 16.6 Building SQL Test Cases

The objective of a SQL Test Case Builder is to capture the information pertaining to a SQL-related problem, along with the exact environment under which the problem occurred, so that the problem can be reproduced and tested on a separate Oracle database instance. Once the test case is ready, you can upload the problem to Oracle Support to enable support personnel to reproduce and troubleshoot the problem.

*The information gathered by SQL Test Case Builder includes the query being executed, table and index definitions (but not the actual data), PL/SQL functions, procedures, and packages, optimizer statistics, and initialization parameter settings.*

# Topic 15: Managing Database Performance

## Capturing Workload Prerequisites

\Workload can be replayed, the logical state of the application data on the replay system should be similar to that of the capture system when replay begins.

While this step is not required, Oracle recommends that the database be restarted before capturing the workload to ensure that ongoing and dependent transactions are allowed to be completed or rolled back before the capture begins.

## Replaying Workload Procedure

1. [Setting Up the Replay Directory](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CHDJBBFI)
2. [Restoring the Database](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CHDBFBCA)
3. [Resolving References to External Systems](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CHDBIBHI)
4. [Remapping Connections](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CHDHHDHI)
5. [User Remapping](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CBAEAIFB)
6. [Specifying Replay Options](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CHDFAGEA)
7. [Using Filters with Workload Replay](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#BABIFBHG)
8. [Setting Up Replay Clients](https://docs.oracle.com/cd/E11882_01/server.112/e41481/dbr_replay.htm#CHDICJHE)

## Something about Automatic Tuning Optimizer

There are two types of the optimizer, a normal mode and tuning mode. Normal mode generates reasonable excution plan for the vast majority of SQL statements. Tuning mode uses additional analysis to check if there is better execution plan than one's generated by normal mode. Normal mode uses very strict time for tuning SQL statement when tuning mode can take several minutes for tuning single statement.

The optimizer perform four types of tuning analysis:

Statistics Analysis: Using object statistics

SQL Profiling: Missing infomation may lead the optimizer to poor excution plan,

Access Path Analysis: the optimizer explires whether a new index can enhance the performance of a query.

SQL Structure Analysis.

## About Automated Maintainance Tasks:

Consist of three advisors:

Automatic Optimizer Statistics Collections

Automatic Segment Advisor

Automatic SQL Tuning Advisor.

## 17.2.1 How Automatic SQL Tuning Works

After automatic SQL tuning begins, the database performs the following steps:

1. Identifies SQL candidates in the AWR for tuning
2. Tunes each SQL statement individually by calling SQL Tuning Advisor

*During the tuning process, the database considers and reports all recommendation types, but it can implement only SQL profiles automatically.*

1. Tests SQL profiles by executing the SQL statement
2. Optionally, implements the SQL profiles provided they meet the criteria of threefold performance improvement

## About SQL Tuning Sets

A [SQL tuning set (STS)](https://docs.oracle.com/database/121/TGSQL/tgsql_glossary.htm#BABHHFCG) is a database object that includes:

* A set of SQL statements
* Associated execution context, such as user schema, application module name and action, list of bind values, and the environment for [SQL compilation](https://docs.oracle.com/database/121/TGSQL/tgsql_glossary.htm#BABJACGD) of the cursor
* Associated basic execution statistics, such as elapsed time, CPU time, buffer gets, disk reads, rows processed, cursor fetches, the number of executions, the number of complete executions, optimizer cost, and the command type
* Associated execution plans and row source statistics for each SQL statement (optional)

## Some thing about SMB:

This shit stores SQL plan baselines and plan history in SYSAUX, also SQL profiles.You can capture plan history automaticly or manually load plan as SQL plan baselines.

SQL plan baselines is set of accepted plan for a SQL statement. These accepted plans have been proven to perform well.

## Capturing DOES not include these following shits:

 Direct path load of data from external files using utilities such as SQL\*Loader

 Non-PL/SQL based Advanced Queuing (AQ)

 Flashback queries

 Oracle Call Interface (OCI) based object navigations

 Non SQL-based object access

 Distributed transactions

 XA transactions

 JAVA\_XA transactions

 Database Resident Connection Pooling (DRCP)

 Workloads using OUT binds

 Multi-threaded Server (MTS) and shared server sessions with synchronization mode set to OBJECT\_ID

 Migrated sessions