ORACLE
University

Integrated Cloud Applications & Platform Services

Oracle Database 12*c*: Oracle Automatic Storage Management Administration

Student Guide

D81242GC20

Edition 2.0 | February 2018 | D103140

Learn more from Oracle University at **education.oracle.com**

ORACLE

Author

Sean Kim

Technical Contributors and Reviewers

Sean Kim

Jerry Lee

Joel Goodman

Frank Fu

James Womack

Graphic Designer

Kavya Bellur

Publishers

Jobi Varghese

Jayanthy Keshavamurthy

Asief Baig

Contents

1

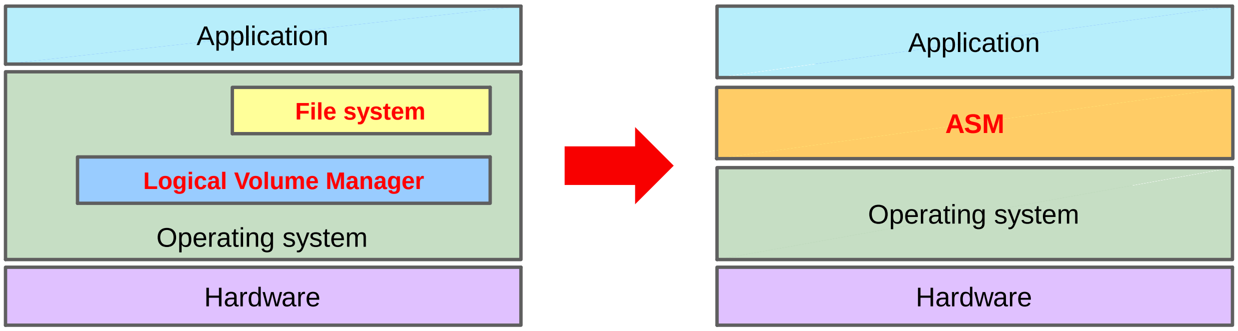
ASM Overview

Objectives

After completing this lesson, you should be able to describe the following:

- Automatic Storage Management (ASM) architecture
- Components of ASM

What Is Oracle ASM?



Oracle ASM is a volume manager and a file system for Oracle Database files that supports single-instance Oracle Database and Oracle Real Application Clusters (Oracle RAC) configurations. Oracle ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices.

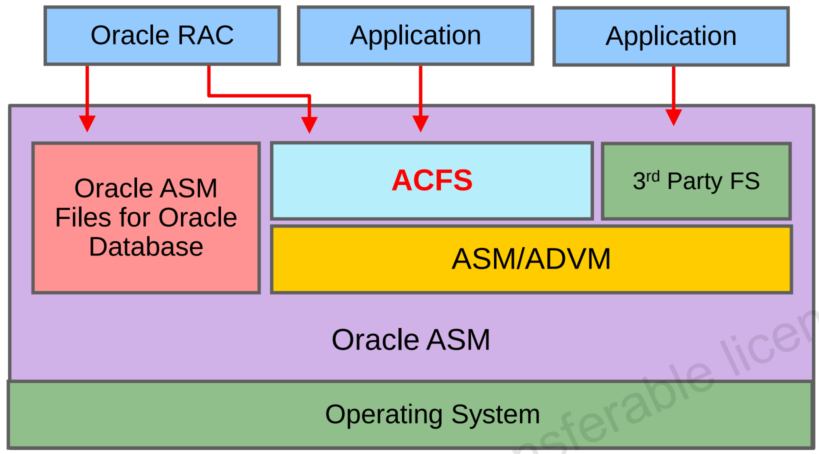
Combining volume management functions with a file system allows a level of integration and efficiency that would not otherwise be possible. For example, ASM is able to avoid the overhead associated with a conventional file system and achieve native raw disk performance for Oracle data files and other file types supported by ASM.

ASM is engineered to operate efficiently in both clustered and nonclustered environments.

Oracle ASM files can coexist with other storage management options such as raw disks and third-party file systems. This capability simplifies the integration of Oracle ASM into pre-existing environments.

Oracle ACFS

- Multi-platform, Scalable file system, and storage management technology that extends Oracle ASM functionality
- Support Oracle database files and other files.
- Spreads data across disks to balance load.
- Provides integrated mirroring across disks.



Oracle Automatic Storage Management Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM) extend Oracle ASM functionality.

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support all customer files. The Oracle ASM Dynamic Volume Manager (Oracle ADVM) provides volume management services and a standard disk device driver interface to clients.

ASM Features and Benefits

- Stripes files rather than logical volumes
- Provides redundancy on a file basis
- Enables online disk reconfiguration and dynamic rebalancing
- Significantly reduces the time to resynchronize a transient failure by tracking changes while the disk is offline
- Provides adjustable rebalancing speed
- Is cluster-aware
- Supports reading from mirrored copy instead of primary copy for extended clusters
- Is automatically installed as part of Grid Infrastructure

ORACLE

ASM provides striping and mirroring without the need to purchase a third-party Logical Volume Manager. ASM divides a file into pieces and spreads them evenly across all the disks. ASM uses an index technique to track the placement of each piece. Traditional striping techniques use mathematical functions to stripe complete logical volumes. ASM is unique in that it applies mirroring on a file basis, rather than on a volume basis. Therefore, the same disk group can contain a combination of files protected by mirroring or not protected at all.

When your storage capacity changes, ASM does not restripe all the data. However, in an online operation, ASM moves data proportional to the amount of storage added or removed to evenly redistribute the files and maintain a balanced I/O load across the disks. You can adjust the speed of rebalance operations to increase or decrease the speed and adjust the impact on the I/O subsystem. This capability also enables fast resynchronization of disks that may suffer a transient failure.

ASM supports all Oracle database file types. It supports Real Application Clusters (RAC) and eliminates the need for a cluster Logical Volume Manager or a cluster file system. In extended clusters, you can set a preferred read copy.

ASM is included in the Grid Infrastructure installation. It is available for both the Enterprise Edition and Standard Edition installations.

ASM Software Installation

- Oracle ASM is installed as part of an Oracle Grid Infrastructure installation either for a standalone cluster or for a standalone server.
- Oracle ASM is installed in the Oracle Grid Infrastructure home before Oracle Database is installed in a separate Oracle home.
- Oracle ASM is integrated with Oracle Clusterware.
- Oracle Clusterware or Oracle Restart is responsible for:
  - Starting and stopping ASM instances
  - Mounting and dismounting disk groups
  - Enabling and disabling ACFS volumes
  - Starting and stopping dependent database instances

ORACLE

Starting with Oracle Database 11*g* Release 2 (11.2), Oracle ASM is installed as part of an Oracle Grid Infrastructure installation either for a standalone cluster or for a standalone server.

ASM depends on the services provided by Oracle Clusterware in both clustered and nonclustered environments. The services allow for cooperating applications, such as the ASM instance and database instances, to communicate in a peer environment and to discover the status of other applications. For ASM, the services:

- Enable the database instance to locate the ASM instance along with the credentials to the ASM instance for establishing an Oracle Call Interface (OCI) connection
- Provide assistance in performing lock recovery. The ASM instance maintains certain locks on behalf of database instances. If a database instance were to fail, ASM uses the group services of `ocssd.bin` to confirm that all database processes are terminated before releasing locks.
- Offer a guarantee that the ASM disk group number that is assigned to a disk group name at run time is unique

Oracle Clusterware is responsible for the startup and shutdown of the ASM instance along with the dependent database instances and resources, as well as the resources on which ASM depends.

Understanding Oracle ASM Concepts

The key components of Oracle ASM include:

- Oracle ASM Instances
- Oracle ASM Disk Groups
- Mirroring and Failure Groups
- Oracle ASM Disks
- Oracle ASM Allocation Units
- Oracle ASM Files

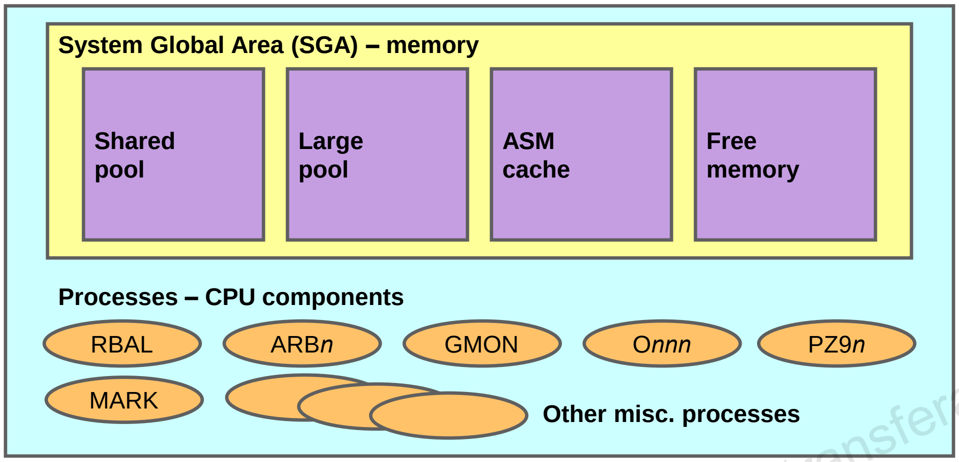
The concepts for the key Oracle ASM components are introduced in the following slides.

- Oracle ASM Instances
- Oracle ASM Disk Groups
- Mirroring and Failure Groups
- Oracle ASM Disks
- Oracle ASM Allocation Units
- Oracle ASM Files

The detailed information of each component is discussed in later lessons.

ASM Components: ASM Instance

The ASM instance comprises the process and memory components for ASM.



ASM instance

Every time ASM or a database is started, a shared memory area called the System Global Area (SGA) is allocated and the ASM background processes are started. However, because ASM performs fewer tasks than a database, an ASM SGA is much smaller than a database SGA. The combination of background processes and the SGA is called an Oracle ASM instance. The instance represents the CPU and RAM components of a running ASM environment.

The SGA in an ASM instance is different in memory allocation and usage than the SGA in a database instance. The SGA in the ASM instance is divided into four primary areas as follows:

- **Shared Pool:** Used for metadata information
- **Large Pool:** Used for parallel operations
- **ASM Cache:** Used for reading and writing blocks during rebalance operations
- **Free Memory:** Unallocated memory available

The minimum recommended amount of memory for an ASM instance is 256 MB. Automatic memory management is enabled by default on an ASM instance and will dynamically tune the sizes of the individual SGA memory components. The amount of memory that is needed for an ASM instance will depend on the amount of disk space that is being managed by ASM.

The second part of the ASM instance is the background processes. An ASM instance can have many background processes; not all are always present.

The background processes that are specific to ASM functionality are covered in the next slide. Because the ASM instance shares the same code base as an Oracle database instance, all the required background processes of a database instance will exist in the ASM instance. There are required background processes and optional background processes. Some of these processes may include the following:

- **ARC*n*:** The archiver processes (exists only when the database is in ARCHIVELOG mode and automatic archiving is enabled)
- **CKPT:** The checkpoint process
- **DBW*n*:** The database writer processes
- **DIAG:** The diagnosability process
- **CJQ0:** The job coordinator process [started and stopped as needed by Oracle Scheduler; dynamically spawns job queue slave processes (**J*nnn***) to run the jobs]
- **J*nnn*:** Job queue processes
- **LGWR:** The log writer process
- **PMON:** The process monitor process
- **PSP0:** The process spawner process
- **QMN*n*:** The queue monitor processes (If `AQ_TM_PROCESSES` is 0, this process will not start.)
- **SMON:** The system monitor process
- **VKTM:** The virtual keeper of time process
- **MMAN:** The memory manager process

The preceding list of processes is not complete. There can be hundreds of database instance background processes running depending on the database options and configuration of the instance. For the ASM instance, these processes will not always perform the same tasks as they would in a database instance. For example, the `LGWR` process in a database instance copies change vectors from the log buffer section of the SGA to the online redo logs on disk. The ASM instance does not contain a log buffer in its SGA, nor does it use online redo logs. The `LGWR` process in an ASM instance copies logging information to an ASM disk group.

If ASM is clustered, additional processes related to cluster management will be running in the ASM instance. Some of these processes include the following:

- **LMON:** The global enqueue service monitor process
- **LMD*n*:** The global enqueue service daemons
- **LMS*n*:** The global cache service processes
- **LCK*n*:** The lock processes

Additional processes are started when ADVM volumes are configured.

- **VDBG:** The Volume Driver Background process forwards ASM requests to lock or unlock an extent for volume operations to the Dynamic Volume Manager driver. The `VDBG` is a fatal background process; termination of this process brings down the ASM instance.
- **VBG*n*:** The Volume Background processes wait for requests from the Dynamic Volume Manager driver, which need to be coordinated with the ASM instance. An example of such a request would be opening or closing an ASM volume file when the Dynamic Volume Manager driver receives an open for a volume (possibly due to a file system mount request) or close for an open volume (possibly due to a file system unmount request), respectively. Unplanned death of any of these processes does not have an effect on the ASM instance.
- **VMB:** Volume Membership Background coordinates cluster membership with the ASM instance.

ASM Instance Primary Processes

The primary processes of an ASM instance are responsible for the ASM-related activities.

Process	Description
RBAL	Opens all device files as part of discovery and coordinates the rebalance activity
ARBn	Refers to one or more slave processes that perform the rebalance activity
GMON	Is responsible for managing disk-level activities such as drop or offline and advancing ASM disk group compatibility
MARK	Marks ASM allocation units as stale following a missed write to an offline disk
Onnn	Represents one or more ASM slave processes that form a pool of connections to the ASM instance for exchanging messages
PZ9n	Represents one or more parallel slave processes that are used for fetching data on clustered ASM installations from GV$ views

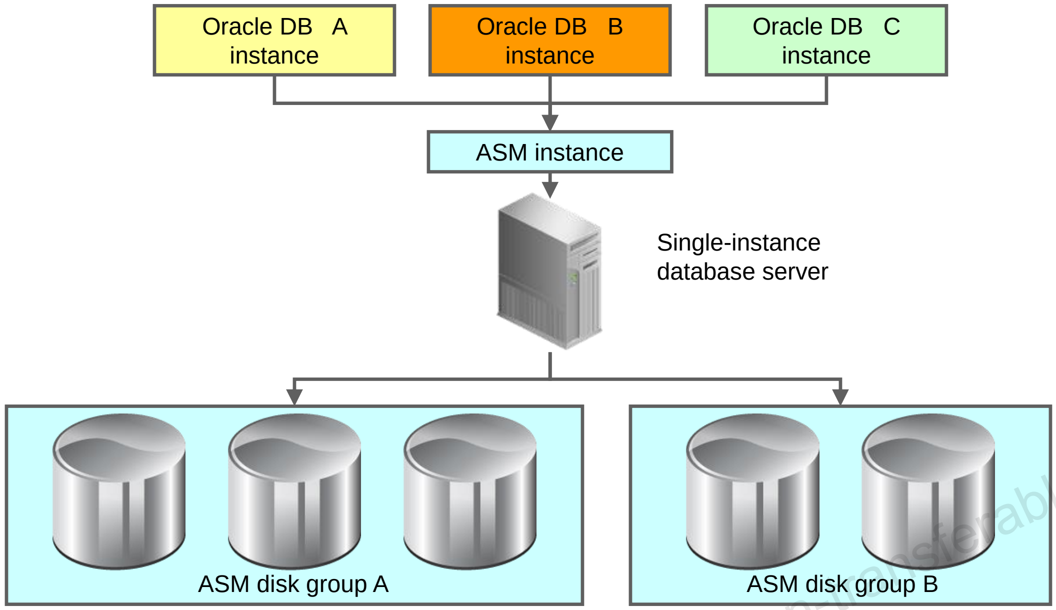
ORACLE

The ASM instance uses dedicated background processes for much of its functionality.

- The RBAL process coordinates the rebalance activity for the disk groups in an ASM instance. It performs a global open on the ASM disks.
- The ARB*n* processes perform the actual rebalance data extent movements in an ASM instance. There can be many of these at a time, called ARB0, ARB1, and so on.
- The GMON process maintains disk membership in the ASM disk groups.
- The MARK process marks ASM allocation units as stale following a missed write to an offline disk.
- The O*nnn* processes represent the server side of a client/server connection. These processes appear the moment the instance is started, and disappear soon after. They form a pool of connections to the ASM instance for exchanging messages and appear only when needed.
- The PZ9*n* processes represent one or more parallel slave processes that are used for fetching data when ASM is running in a clustered configuration on more than one machine concurrently.

ASM Instance Designs:

***Case 1:** Non-clustered ASM and Oracle Databases*

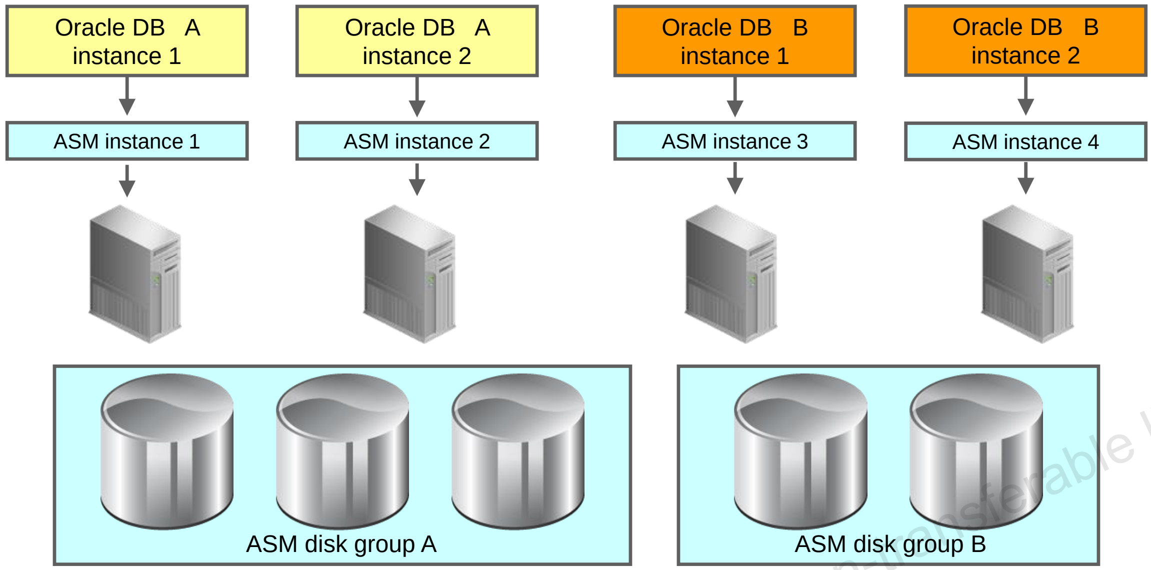


The slide illustrates the first of three ASM instance designs. This design uses a nonclustered ASM environment for one or more nonclustered databases. At the bottom of the graphic, there are two disk groups that the ASM instance uses to provide space for the databases. In this environment, a single host machine contains the ASM instance along with each of the three database instances. Each database can store its files in both disk groups, or only a single disk group if desired. The ASM instance manages the metadata and provides space allocations for the ASM files that are created by each database. When a database instance creates or opens an ASM file, it communicates those requests to the ASM instance. In response, the ASM instance provides file extent map information to the database instance. The ASM instance does not process actual file I/O. This design is useful for storage consolidation.

Note: Refer to MOS Note: 1584742.1 - *Withdrawn: Deprecation Announcement of Oracle Restart with Oracle Database 12.*

ASM Instance Designs:

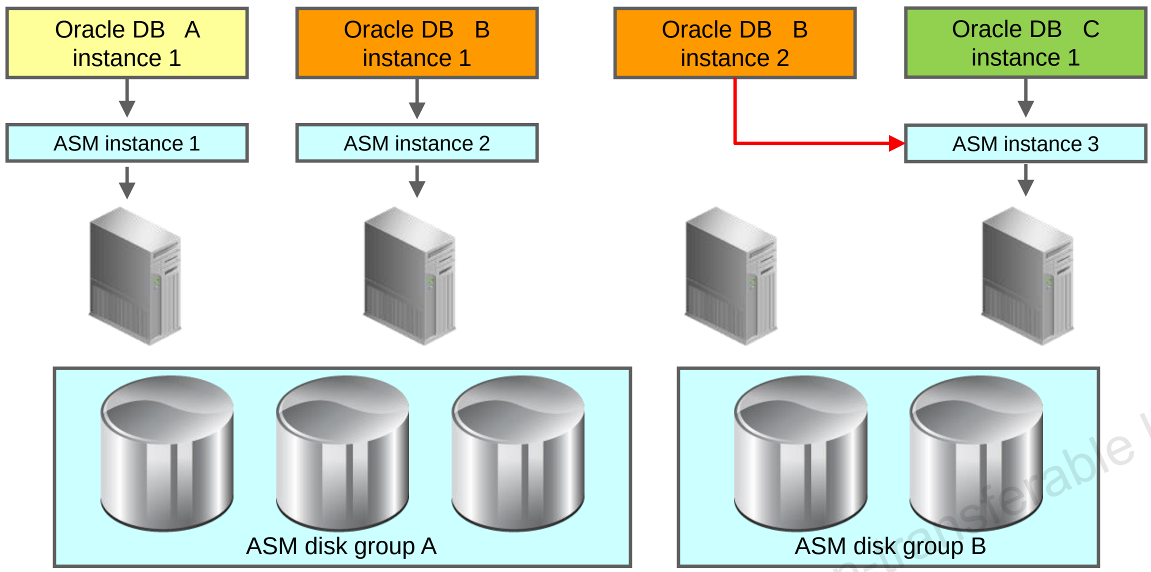
***Case 2:** Clustered ASM for Clustered Databases*



The slide illustrates the second of three ASM instance designs. It shows an ASM cluster in an Oracle RAC database environment where ASM provides a clustered pool of storage. If standard ASM is configured, there is one ASM instance for each node that can provide space management for multiple Oracle RAC or single-instance databases in the cluster. If Flex ASM is configured, it is not necessary for each node that is hosting a database instance to also host an ASM instance. In the example in the slide, two RAC databases are shown: Oracle RAC database “A” has instances on the first two nodes, and Oracle RAC database “B” has instances on the other two nodes. This represents two distinct RAC databases. The ASM instances coordinate with each other when provisioning disk space from the two disk groups that are shown. In this design, ASM provides storage consolidation, and RAC provides database consolidation.

ASM Instance Designs:

Case 3: *Clustered ASM for Mixed Databases*



The slide illustrates the third of three ASM instance designs. This design shows a three-node ASM clustered design that provides a clustered storage pool consisting of disk group “A” and disk group “B.” A shared ASM storage pool is achieved by using Oracle Clusterware. However, in this design, clustered and nonclustered database environments are used on each of the three nodes. This design is useful in a grid environment where any instance can be moved to any database server node, and multiple instances can reside on any node.

In the example in the slide, Oracle Database C instance 1 is a client of ASM instance 2, which has incorporated Flex ASM. However, the model is still relevant for clusters based on standard ASM.

Note: In Oracle Clusterware 12*c* release 2 (12.2), all standalone clusters are configured as Oracle Flex Clusters, meaning that a cluster is configured with one or more Hub Nodes and can support a large number of Leaf Nodes. Clusters currently configured under older versions of Oracle Clusterware are converted in place as part of the upgrade process, including the activation of Oracle Flex ASM (which is a requirement for Oracle Flex Clusters).

ASM Components: ASM Disk Group

The ASM disk group is the fundamental object that ASM manages. It:

- Consists of one or more ASM disks that provide space
- Includes self-contained metadata and logging information for management of space within each disk group
- Is the basis for storage of ASM files
- Supports three disk group redundancy levels:
  - Normal defaults to internal two-way mirroring of ASM files.
  - High defaults to internal three-way mirroring of ASM files.
  - External uses no ASM mirroring and relies on external disk hardware or redundant array of inexpensive disks (RAID) to provide redundancy.
- Supports ASM files from multiple databases

The ASM disk group consists of one or more ASM disks that provide space, and is the fundamental object that ASM manages. Each disk group contains its own metadata and logging information that is required for management of space within that disk group. Files are allocated from disk groups. Any Oracle ASM file is completely contained within a single disk group. However, a disk group might contain files that belong to several databases and a single database can use files from multiple disk groups. For most installations, you need only a small number of disk groups, usually two, and rarely more than three.

Mirroring protects data integrity by storing copies of data on multiple disks. When you create a disk group, you specify an Oracle ASM disk group type based on one of the following three redundancy levels:

- `Normal` for two-way mirroring
- `High` for three-way mirroring
- `External` to not use Oracle ASM mirroring, such as when you configure hardware RAID for redundancy

The redundancy level controls how many disk failures are tolerated without dismounting the disk group or losing data. The disk group type determines the mirroring levels with which Oracle creates files in a disk group.

ASM Components: Failure Groups

A failure group is a subset of the disks in a disk group, which could fail at the same time because of shared hardware.

- Failure groups enable mirroring of metadata and user data.
- Default failure group creation puts every disk in its own failure group.
- Multiple disks can be placed in a singe failure group.
- Failure groups apply only to normal and high redundancy disk groups.
  - A normal redundancy disk group requires at least two failure groups to implement two-way mirroring of files.
  - A high redundancy disk group requires at least three failure groups to implement three-way mirroring of files.
- A quorum failure group contains copies of voting files when they are stored in normal or high redundancy disk groups.

ORACLE

A failure group is a subset of the ASM disks in a disk group, which could fail at the same time because of shared hardware. The failure of common hardware must be tolerated. Four drives that are in a single removable tray, not protected with RAID, should be in the same failure group because the tray could be removed, thus making all four drives fail at the same time. Drives in the same cabinet could be in multiple failure groups if the cabinet has redundant power and cooling so that it is not necessary to protect against failure of the entire cabinet.

Failure groups are used to store mirror copies of data when ASM is used for mirroring by declaring the disk group redundancy to be Normal or High at creation time. A normal redundancy disk group requires at least two failure groups to implement two-way mirroring of files. A high redundancy disk group requires at least three failure groups to implement three-way mirroring of files.

There are always failure groups even if they are not explicitly created. If you do not specify a failure group for a disk, that disk is placed in its own failure group with a failure group name that is the same as the disk name. Therefore, if 20 disks were in a single disk group, there could be 20 failure groups as well. Failure groups have meaning only when they are used with normal and high redundancy disk groups. All failure groups within the same disk group should be created with the same capacity to avoid space allocation problems.

A quorum failure group is a special type of failure group that contains mirror copies of voting files when voting files are stored in normal or high redundancy disk groups.

ASM Components: ASM Disks

ASM disks are the storage devices that are provisioned to ASM disk groups.

- Are formed from five sources as follows:
  - A disk or partition from a storage array
  - An entire physical disk or partitions of a physical disk
  - Logical volumes (LV) or logical units (LUN)
  - Network-Attached Files (NFS)
  - Exadata grid disk
- Are named when they are added to a disk group by using a different name than the operating system device name
- May use different operating system device names on different nodes in a cluster for the same ASM disk
- Are divided into allocation units (AU) with sizes 1, 2, 4, 8, 16, 32, or 64 MB allowed

ORACLE

ASM disks are the storage devices that provide space to ASM disk groups. They are not always the same as physical disks. There are five sources for an ASM disk as follows:

- A disk or partition from a storage array
- An entire physical disk or individual partitions of a physical disk
- Logical volumes (LV) or logical units (LUN)
- Network-Attached Files (NFS)
- Exadata grid disk

ASM uses the ASM disk name, not the OS name, that is assigned to the disk when it is added to the ASM disk group. ASM disk names allow a common logical naming convention to be used. Every ASM disk is divided into allocation units (AUs). An AU is the fundamental unit of allocation within a disk group. A file extent consists of one or more AUs. An ASM file consists of one or more file extents. Each file extent is allocated to a single ASM disk. When you create a disk group, you can set the 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 disk group compatibility level. Larger AU sizes typically provide performance advantages for data warehouse applications that use large sequential reads. ASM spreads files evenly across all the disks in the disk group. This allocation pattern maintains every disk at the same capacity level and the same I/O load. Different ASM disks should not share the same physical drive.

ASM Components: ASM Files

ASM files are a limited set of file types stored in an ASM disk group.

Some supported file types:

Control files	Flashback logs	Data Pump dump sets
Data files	DB SPFILE	Data Guard configuration
Temporary data files	RMAN backup sets	Change tracking bitmaps
Online redo and archive logs	RMAN data file copies	Disaster recovery configurations
ASM and DB password files	Transport data files	ASM SPFILE

- Are stored as a set or collection of data extents
- Are striped across all disks in a disk group
- Use names that begin with a plus sign (+), which are automatically generated or from user-defined aliases

ORACLE

ASM files are a limited set of file types stored in an ASM disk group. These files are not created directly by ASM, but ASM automatically generates the file names. ASM file names begin with a plus sign (+) followed by a disk group name. In addition, the file name will contain the name of the database that created it along with a file type qualifier name. A fully qualified file name has the following form:

```
+group/dbname/file_type/file_type_tag.file.incarnation
```

You can think of the plus sign (+) as the root directory of the ASM file system, similar to the slash (/) on Linux file systems. An example of a fully qualified ASM file name is:

```
+dgroup2/prod/controlfile/current.256.541956473
```

The chart in the slide lists the valid types of files that can be stored in an ASM disk group. Each ASM file must be contained within a single disk group, but a single Oracle database can have files in multiple disk groups. You can specify user-friendly aliases for ASM files by creating a hierarchical directory structure and use directory names as prefixes to the file names.

ASM Files: Extents and Striping

ASM can use variable size data extents to support larger files, reduce memory requirements, and improve performance.

- Each data extent resides on an individual disk.
- Data extents consist of one or more allocation units.
- For disk groups with AU size less than 4MB:
  - Equal to AU for the first 20,000 extents (0–19999)
  - Equal to 4 × AU for the next 20,000 extents (20000–39999)
  - Equal to 16 × AU for extents above 40,000

ASM stripes files by using extents with a coarse method for load balancing or a fine method to reduce latency.

- Coarse-grained striping is always equal to the effective AU size.
- Fine-grained striping is always equal to 128 KB.

ORACLE

The contents of ASM files are stored in a disk group as a set, or a collection, of data extents that are stored on individual disks within disk groups. Each extent resides on an individual disk. Extents consist of one or more AUs. To accommodate increasingly larger files, ASM uses variable size extents.

Variable size extents enable support for larger Oracle ASM data files, reduce SGA memory requirements for very large databases, and improve performance for file create and open operations. The initial extent size is equal to the allocation unit size and it increases by a factor of 4 or 16 at predefined thresholds. The extent size increases by a factor of 4 after 20,000 extents, and by a factor of 16 after 40,000 extents, where it remains at 16 × AU size.

ASM uses coarse-grained striping to balance loads across all the disks in a disk group and fine-grained striping to reduce I/O latency. Coarse-grained stripes are always equal to the effective AU size. Effective AU size is the AU size defined by a disk group at creation. It can vary for each disk group. The fine-grained stripe size always equals 128 KB. A fine-grained stripe is taken from within a series of coarse-grained stripes. Control files use fine-grained striping by default. All other file types use coarse-grained striping. This default striping method of coarse-grained or fine-grained can be changed with custom templates.

ASM Files: ASM File Mirroring

ASM mirroring is specified at the file level.

- Two files can share the same disk group with one file being mirrored while the other is not.
- ASM allocates extents for a file with the primary and mirrored copies in different failure groups.
- The mirroring options for ASM disk group types are:

Disk Group Type	Supported Mirroring Levels	Default Mirroring Level
External redundancy	Unprotected (None)	Unprotected (None)
Normal redundancy	Two-way Three-way Unprotected (None)	Two-way
High redundancy	Three-way	Three-way

ORACLE

Mirroring protects data integrity by storing copies of data on multiple disks that are isolated from a single failure. ASM mirroring is more flexible than traditional RAID mirroring because you can specify the redundancy level for each file within a disk group. Two files can share the same disk group with one file being mirrored while the other is not.

When ASM allocates an extent for a normal redundancy file (two-way mirroring), it allocates a primary extent and a secondary extent. ASM chooses the disk on which to store the secondary extent in a different failure group other than the primary extent. Therefore, simultaneous failure of all disks in a failure group does not result in data loss.

The table in the slide lists the default disk group, the supported mirroring levels for the files within that disk group, and the default mirroring level for any file created in the disk group unless a custom mirroring level is designated.

With an external redundancy disk group, ASM relies on the storage system to provide RAID functionality. Any write errors will cause a forced dismount of the entire disk group. With normal redundancy, a loss of one ASM disk is tolerated. With high redundancy, a loss of two ASM disks in different failure groups is tolerated.

The REQUIRED_MIRROR_FREE_MB column of V$ASM_DISKGROUP indicates the amount of space that must be available in a disk group to restore full redundancy after the worst failure that can be tolerated by the disk group without adding additional storage.

This requirement ensures that there are sufficient failure groups to restore redundancy. Also, worst failure refers to a permanent failure where the disks must be dropped, not the case where the disks go offline, and then come back online.

The amount of space displayed in this column takes the effects of mirroring into account. The value is computed as follows:

- Normal redundancy disk group with more than two failure groups
  The value is the total raw space for all the disks in the largest failure group. The largest failure group is the one with the largest total raw capacity. For example, if each disk is in its own failure group, the value would be the size of the largest capacity disk.
- High redundancy disk group with more than three failure groups

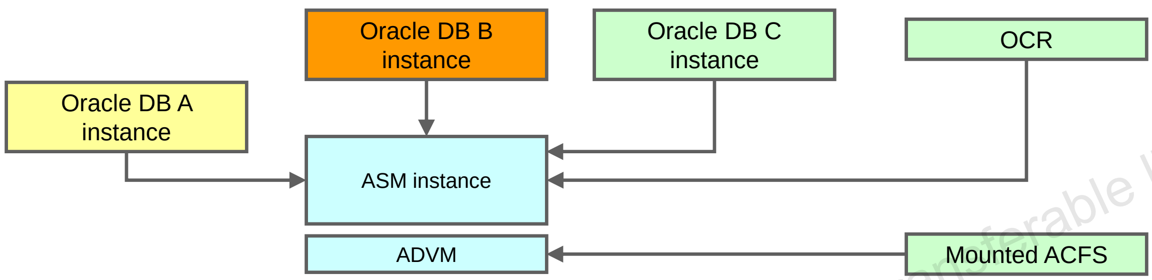
  The value is the total raw space for all the disks in the two largest failure groups.

The `USABLE_FILE_MB` column of `REQUIRED_MIRROR_FREE_MB` indicates the amount of free space, adjusted for mirroring, that is available for new files to restore redundancy after a disk failure. Calculate `USABLE_FILE_MB` by subtracting `REQUIRED_MIRROR_FREE_MB` from the total free space in the disk group, and then adjusting the value for mirroring. For example, in a normal redundancy disk group where, by default, the mirrored files use disk space that is twice their size, if 4 GB of actual usable file space remains, `USABLE_FILE_MB` equals roughly 2 GB. You can then add a file that is up to 2 GB.

ASM Clients

Any active database instance that is using ASM storage and is currently connected to the ASM instance is an ASM client.

- Connected ASM clients can be viewed:
  - By using the `asmcmd lsct disk_group` command
  - In the `v$asm_client` dynamic performance view
- Each file in ASM is associated with a single database.



Any active database instance that is using ASM storage and is currently connected to the ASM instance is an ASM client. Each file in ASM is associated with only one database or client at a time (excepting OCR and voting disk files). This can be observed with the fully qualified ASM file name. The second field of the fully qualified ASM file name is the database name as follows:

```
+group/dbname/file_type/file_type_tag.file.incarnation
```

ASM clients are tracked in the v$asm_client dynamic performance view. There will exist one row for each combination of distinct database instance name and disk group number that is being used.

An OCR file and a voting file that are stored in ASM are listed as an ASM client with the name +ASM.

ADVM volumes are ASM clients and the file system must be dismounted before the instance can be shut down. ASM volumes have a client name of asmvol.

Note: Many utilities are capable of connecting to the ASM instance to perform administration. These utilities are sometimes referred to as ASM clients, but they do not appear in the v$asm_client dynamic performance view. These are referred to as ASM utilities in this course.

ASM Utilities

Many utilities can be used for ASM administration. These utilities may include:

- Oracle Universal Installer (OUI)
- ASM Configuration Assistant (ASMCA)
- Oracle Enterprise Manager (EM)
- SQL*Plus
- ASM Command-Line utility (ASMCMD)
- Server controller utility (srvctl)

ORACLE

There are many utilities that can be used for ASM administration. Some of these utilities connect directly to the ASM instance and others can scan the ASM disks headers directly. The utilities may include:

- **Oracle Universal Installer (OUI):** It is used to install the ASM software and can create the initial disk groups.
- **ASM Configuration Assistant (ASMCA):** It is used to initially configure the ASM instance and create disk groups. It can be invoked from within the OUI utility.
- **Oracle Enterprise Manager (EM):** It is used to perform central administration of a grid environment, including the ASM instance by using a graphical client interface.
- **SQL*Plus:** It is used to provide command-line SQL language access to the ASM instance. All ASM commands and administration can be performed with this utility.
- **ASM Command-Line utility (ASMCMD):** It is used for ASM administration from the command line without the SQL language. It uses the UNIX style syntax.
- **Server controller utility:** It is used to start, stop, and check the status of all the ASM instances in a cluster environment with a single command.

Quiz

Select the utilities that can be used to configure or manage ASM.

a. ASM Configuration Assistant (ASMCA)
b. Oracle Enterprise Manager (EM)
c. Database Configuration Assistant (DBCA)
d. SQL*Plus
e. ASM Command-Line utility (ASMCMD)

Answer: a, b, d, e

Summary

In this lesson, you should have learned how to describe the following:

- Automatic Storage Management (ASM) architecture
- Components of ASM

2

Administering ASM Instances

Objectives

After completing this lesson, you should be able to:

- Explain and apply Automatic Storage Management (ASM) initialization parameters
- Manage ASM instances and associated processes
- Monitor ASM

ORACLE

Operating with Different Releases of Oracle ASM and Database Instances Simultaneously

- Oracle Automatic Storage Management (Oracle ASM) in Oracle Database 12*c* Release 2 (12.2) supports Oracle Database 12*c* Release 2 (12.2) or older software versions, including Oracle Database 10*g* Release 1 (10.1).
- When using different software versions, the database instance supports Oracle ASM functionality of the earliest release in use.
  - For example, an Oracle Database 10*g* Release 1 (10.1) database instance operating with an Oracle ASM 12*c* Release 2 (12.2) instance only supports Oracle ASM 10*g* Release 1 (10.1) features.

Oracle Automatic Storage Management (Oracle ASM) in Oracle Database 12*c* Release 2 (12.2) supports Oracle Database 12*c* Release 2 (12.2) or older software versions, including Oracle Database 10*g* Release 1 (10.1). An Oracle ASM instance must be at Oracle ASM 12*c* Release 2 (12.2) to support Oracle Database 12*c* Release 2 (12.2). There are additional compatibility considerations when using disk groups with different releases of Oracle ASM and database instances.

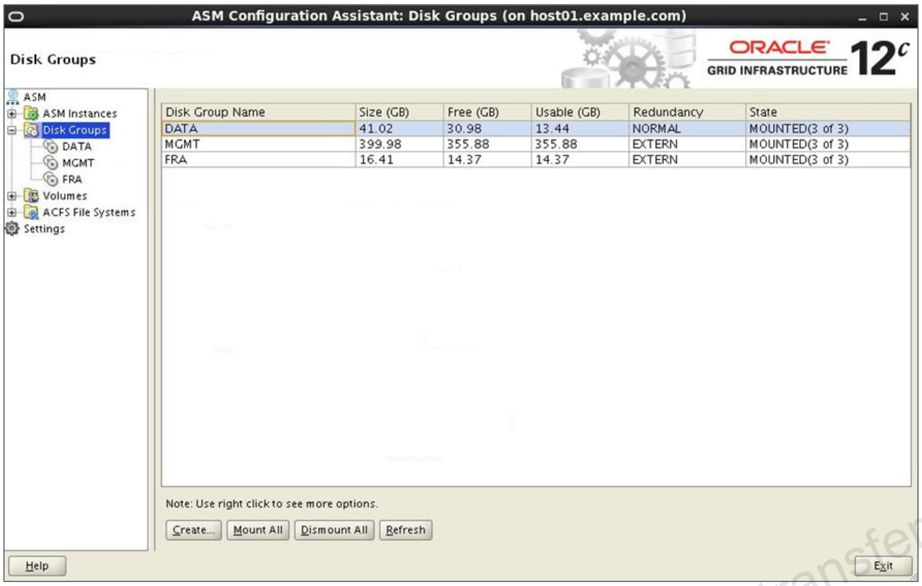
When using different software versions, the database instance supports Oracle ASM functionality of the earliest release in use. For example, an Oracle Database 10*g* Release 1 (10.1) database instance operating with an Oracle ASM 12*c* Release 2 (12.2) instance only supports Oracle ASM 10*g* Release 1 (10.1) features.

The `V$ASM_CLIENT` view contains the `SOFTWARE_VERSION` and `COMPATIBLE_VERSION` columns with information about the software version number and instance compatibility level.

- The `SOFTWARE_VERSION` column of `V$ASM_CLIENT` contains the software version number of the database or Oracle ASM instance for the selected disk group connection.
- The `COMPATIBLE_VERSION` column contains the setting of the `COMPATIBLE` parameter of the database or Oracle ASM instance for the selected disk group connection.

You can query the `V$ASM_CLIENT` view on both Oracle ASM and database instances.

Managing ASM with ASMCA



ORACLE

The ASM Configuration Assistant (ASMCA), ASMCMD, and SQL*Plus are located in the `bin` directory of the Grid Infrastructure home. All these utilities require that the `ORACLE_SID` environment variable be set. By using the `oraenv` script, you can set the `ORACLE_SID`, `ORACLE_HOME`, and `PATH` variables.

An example is as follows, assuming that you are connected as the `grid` software owner on the first node of your cluster:

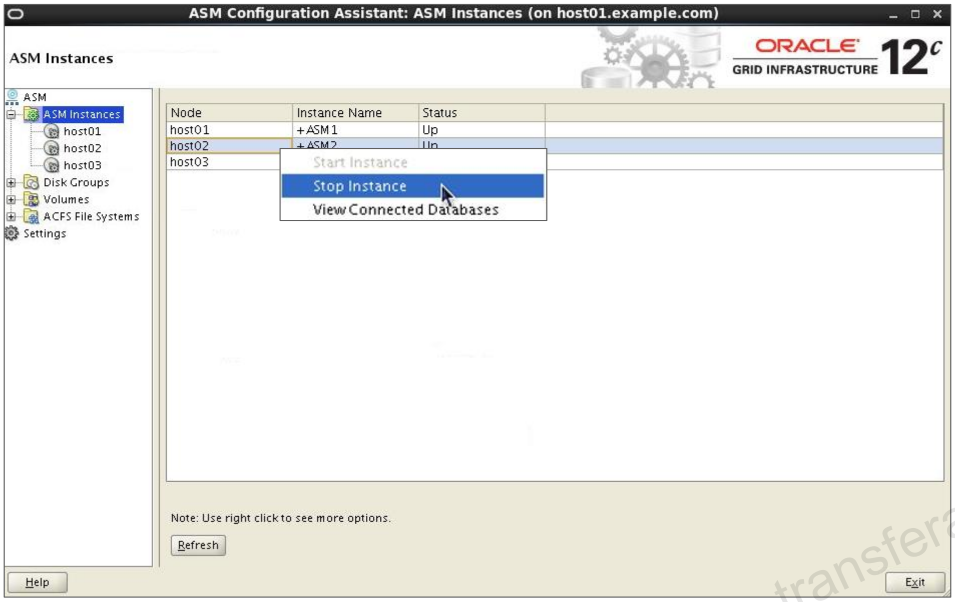
```
$ . oraenv
ORACLE_SID = [+ASM1] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/12.2.0/grid is /u01/app/grid
```

Start the ASM Configuration Assistant (ASMCA) utility with:

```
$ asmca
```

ASMCA provides a complete set of commands to manage the ASM instances, disk groups, volumes, and ASM Cluster File Systems.

Starting and Stopping ASM Instances by Using ASMCA



The ASMCA utility that is shown in the slide allows you to start and stop an ASM instance.

Starting and Stopping ASM Instances by Using ASMCMD

- Use ASMCMD to shut down an Oracle ASM instance.
  - Shut down normally:
    ```
    ASMCMD> shutdown [--normal]
    ```
  - Shut down after aborting current operations:
    ```
    ASMCMD> shutdown [--abort]
    ```
  - Shut down immediately:
    ```
    ASMCMD> shutdown [--immediate]
    ```
- Use ASMCMD to start an Oracle ASM instance.
  - Start instance normally:
    ```
    ASMCMD> startup
    ```
  - Start by using the `asm_init.ora` parameter file:
    ```
    ASMCMD> startup --pfile asm_init.ora
    ```
  - Start in restricted mode:
    ```
    ASMCMD> startup --restrict
    ```


The ASMCMD utility includes the ability to start and stop ASM instances. The following are examples of the `shutdown` command. The first example in the slide performs a shutdown of the Oracle ASM instance with normal action. The second example performs a shutdown with immediate action. The third example performs a shutdown that aborts all existing operations.

```
$ asmcmd
ASMCMD [+] > shutdown  (or shutdown --normal)
ASMCMD [+] > shutdown --immediate
ASMCMD [+] > shutdown --abort
```

Following are examples of ASMCMD startup commands. The `--nomount` option specifies the nomount operation and `--restrict` specifies restricted mode.

```
ASMCMD> startup --nomount
ASMCMD> startup --restrict
```

The following is an example of the `startup` command that starts the Oracle ASM instance without mounting disk groups and uses the `asm_init.ora` initialization parameter file.

```
ASMCMD> startup --nomount --pfile asm_init.ora
```

Starting and Stopping ASM Instances by Using `srvctl`

The Server Control utility (`srvctl`) can be used to start and stop ASM instances.

- One node at a time:

```
$ srvctl start asm -n host01
$ srvctl start asm -n host02
$ srvctl status asm -n host01
ASM is running on host01.
$ srvctl status asm -n host02
ASM is running on host02.
```

- All nodes simultaneously:

```
$ srvctl stop asm -f
$ srvctl status asm -n host01
ASM is not running on host01.
$ srvctl status asm
ASM is not running on host01,host02.
```


The `srvctl` utility can be used to start and stop ASM instances along with other resources managed by Oracle Grid Infrastructure. `srvctl` can be found under the *Grid_home*/`bin` location that is established during Oracle Grid Infrastructure installation. One of the major advantages of `srvctl` is that it allows you to easily start or stop all the ASM instances in your cluster from any node in the cluster. `srvctl` can be used to control ASM in the following ways:

- Start an ASM instance.

  `srvctl start asm [-n <node>] [-o <start_option>]`

  <node> is the name of the cluster node that hosts the ASM instance (optional); if omitted, the ASM instance is started on all nodes.

  <asm_inst> is the name of the specific ASM instance that is being acted upon (optional).

  <start_option> is one of the valid instance startup options (`FORCE`, `MOUNT`, `OPEN`, `NOMOUNT`, or `RESTRICT`) (optional).

- Stop an ASM instance.

  `srvctl stop asm [-n <node>] [-o <stop_option>]`

  <stop_option> is one of the valid instance shutdown options (`NORMAL`, `IMMEDIATE`, `TRANSACTIONAL`, or `ABORT`) (optional).

- Report the status of an ASM instance.

  `srvctl status asm [-n <node>]`

Starting and Stopping ASM Instances By Using SQL*Plus

Starting and stopping ASM instances by using SQL*Plus is similar to how you start and stop database instances.

```
$ export ORACLE_SID=+ASM1
$ export ORACLE_HOME=/u01/app/12.2.0/grid
$ $ORACLE_HOME/bin/sqlplus / AS SYSASM

SQL*Plus: Release 12.2.0.1.0 Production on Tue Jul 30 19:56:51 2016


Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
With the Real Application Clusters and ASM options
SQL> startup
ASM instance started

Total System Global Area  284565504 bytes
Fixed Size                  1312896 bytes
Variable Size             258086784 bytes
ASM Cache                  25165824 bytes
ASM diskgroups mounted

SQL>
```


With SQL*Plus, you start an ASM instance by using the `STARTUP` command, which is similar to how you start an Oracle Database instance. When starting an ASM instance, note the following:

- To connect to an ASM instance with SQL*Plus, set the `ORACLE_SID` environment variable to the ASM SID. The default ASM SID for a single instance database is `+ASM` and the default SID for ASM for an Oracle RAC node is `+ASM`*`node_number`*, where *`node_number`* is the number of the node.
- Set the `ORACLE_HOME` environment variable to the ASM installation location.
- The initialization parameter file must contain the following entry:

  `INSTANCE_TYPE = ASM`

  This parameter indicates that an ASM instance, not a database instance, is starting.
- 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`, and the rules for mounting disk groups. If no disk groups meet the rules, the ASM instance starts and ASM displays a message that no disk groups were mounted. You can later mount disk groups with the `ALTER DISKGROUP...MOUNT` command.

  Note: When using Grid infrastructure, CRS is responsible for mounting ASM disk groups.

The following list describes the STARTUP command parameters that are relevant to ASM.

- **FORCE:** Issues a SHUTDOWN ABORT to the ASM instance before restarting it
- **MOUNT or OPEN:** Mounts the disk groups specified in the ASM_DISKGROUPS initialization parameter. In an ASM instance, MOUNT and OPEN are synonymous. OPEN is the default if no command parameter is specified. Again, in a clustered environment, the ASM_DISKGROUPS parameter is not used.
- **NOMOUNT:** Starts up the ASM instance without mounting any disk groups
- **RESTRICT:** Starts up an instance in restricted mode. The RESTRICT clause can be used in combination with the MOUNT, NOMOUNT, and OPEN clauses.
  In restricted mode, database instances cannot use the disk groups. That is, databases cannot open the files that are in that disk group. Also, if a disk group is mounted by an instance in restricted mode, this disk group cannot be mounted by any other instance in the cluster. Restricted mode enables you to perform maintenance tasks on a disk group without interference from clients. The rebalance operations that occur while a disk group is in restricted mode eliminate the lock and unlock extent map messaging that occurs between the ASM instances in a clustered environment. This improves the overall rebalance throughput. At the end of a maintenance period, you must explicitly dismount the disk group and remount it in normal mode.

The ASM shutdown process is initiated when you run the SHUTDOWN command in SQL*Plus. Before you run this command, ensure that the ORACLE_SID and ORACLE_HOME environment variables are set so that you can connect to the ASM instance.

Oracle strongly recommends that you shut down all the database instances that use the ASM instance before attempting to shut down the ASM instance.

The following list describes the SHUTDOWN command parameters that are relevant to ASM.

- **NORMAL:** ASM waits for any in-progress SQL to complete before dismounting all the disk groups and shutting down the ASM instance. Before the instance is shut down, ASM waits for all the currently connected users to disconnect from the instance. If any database instances are connected to the ASM instance, the SHUTDOWN command returns an error and leaves the ASM instance running. NORMAL is the default shutdown mode.
- **IMMEDIATE or TRANSACTIONAL:** ASM waits for any in-progress SQL to complete before dismounting all 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, the SHUTDOWN command returns an error and leaves the ASM instance running.
- **ABORT:** 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, the database instance aborts. (If Flex ASM is configured, however, the database is connected to another Flex ASM instance.)

  If any Oracle Automatic Storage Management Cluster File System (Oracle ACFS) file systems are currently mounted on the Oracle ASM Dynamic Volume Manager (ADVM) volumes, those file systems should first be dismounted.

Starting and Stopping ASM Instances

If Oracle Cluster Registry (OCR) or voting files are stored in a disk group:

- The Oracle ASM instances are automatically restarted by the high availability services daemon
- Use `crsctl stop crs` to shutdown the Oracle ASM instances

ORACLE

If Oracle Cluster Registry (OCR) or voting files are stored in a disk group, the Oracle ASM instances are automatically restarted by the high availability services daemon. The disk group can only be dismounted by shutting down the Oracle ASM instance as part of shutting down the clusterware on a node. To shut down the clusterware, run crsctl stop crs.

ASM Initialization Parameters

- ASM initialization parameters can be set:
  - With Oracle ASM Configuration Assistant (ASMCA)
  - After installation, by using the `ALTER SYSTEM` or `ALTER SESSION` SQL statements
- `INSTANCE_TYPE=ASM` is the only mandatory parameter setting.
- There are a number of ASM-specific parameters.
  - These have names starting with `ASM_`.
- Some database parameters are valid for ASM.
  - Example: `MEMORY_TARGET`
- You can use a `PFILE` or `SPFILE` to manage parameters.

You use Enterprise Manager or the `ALTER SYSTEM` or `ALTER SESSION` SQL statements to change the initialization parameters for an ASM instance after installation.

There are three groupings of parameters for an ASM instance: mandatory, ASM-only, and parameters that are also valid for a database.

- `INSTANCE_TYPE` is the only mandatory parameter for ASM.
- ASM-only parameters have names that are prefixed with `ASM_`. These have suitable defaults for most environments.
- Some database initialization parameters are also valid for ASM (for example, `MEMORY_TARGET`). In general, ASM uses appropriate defaults for relevant database parameters.

Oracle strongly recommends that you use a server parameter file (SPFILE) as the ASM instance parameter file. SPFILE is maintained by the instance, so it must be in a shared location. In a clustered ASM environment, SPFILE is placed by default in an ASM disk group; it could also be located on a cluster file system. You can use a text initialization parameter file (PFILE) on each cluster node, but these must be maintained manually and kept synchronized.

The rules for file name, default location, and search order that apply to database initialization parameter files also apply to ASM initialization parameter files. The search order is: location set in the Grid Plug and Play profile, then for an SPFILE in the Oracle home, and finally for a PFILE in the Oracle home.

ASM_DISKGROUPS

ASM_DISKGROUPS specifies a list of disk group names that ASM automatically mounts at instance startup.

- Uses the default value of NULL
- Is ignored in certain circumstances:
  - If ASM is started with the NOMOUNT option
  - If all disk groups are explicitly mounted
- Can be set dynamically by using ALTER SYSTEM
- Is automatically modified when disk groups are added, deleted, mounted, or unmounted if using an SPFILE
- Must be manually adjusted if using a PFILE
  - Except when ASMCA is used to create a new disk group

The ASM_DISKGROUPS initialization parameter specifies a list of disk group names that ASM automatically mounts at instance startup. The default value of the ASM_DISKGROUPS parameter is a NULL string.

ASM_DISKGROUPS is ignored when you specify the NOMOUNT option at instance startup or when you issue the ALTER DISKGROUP ALL MOUNT statement.

The ASM_DISKGROUPS parameter is dynamic. The following is an example of setting the ASM_DISKGROUPS parameter dynamically:

```
ALTER SYSTEM SET ASM_DISKGROUPS = DATA, FRA;
```

When an SPFILE is used, ASM_DISKGROUPS is modified when a disk group is mounted, dismounted, created, or dropped. When you use ASMCMD or SQL*Plus to mount a disk, the disk is mounted only on the local node and ASM_DISKGROUPS is modified only for the local instance. When you use ASMCA to mount a disk group, you have a choice to mount locally or on all nodes. ASM_DISKGROUPS is modified in accordance with your choice.

When using a PFILE, you must edit the initialization parameter file to add or remove disk group names. The following is an example of the ASM_DISKGROUPS parameter in a PFILE:

```
ASM_DISKGROUPS = DATA, FRA
```

In a clustered environment, this parameter does not need to be set because each disk group is a CRS resource and mounting is managed by Clusterware.

Disk Groups Mounted at Startup

At startup, the Oracle ASM instance attempts to mount the following disk groups:

- Disk groups specified in the `ASM_DISKGROUPS` initialization parameter
- Disk group used by Cluster Synchronization Services (CSS) for voting files
- Disk groups used by Oracle Clusterware for the Oracle Cluster Registry (OCR)
- Disk group used by the Oracle ASM instance to store the ASM server parameter file (SPFILE)

At startup, the Oracle ASM instance attempts to mount the following disk groups:

- Disk groups specified in the `ASM_DISKGROUPS` initialization parameter
- Disk group used by Cluster Synchronization Services (CSS) for voting files
- Disk groups used by Oracle Clusterware for the Oracle Cluster Registry (OCR)
- Disk group used by the Oracle ASM instance to store the ASM server parameter file (SPFILE)

If no disk groups are found in the previous list, the Oracle ASM instance does not mount any disk groups at startup.

ASM_DISKSTRING

ASM_DISKSTRING specifies a list of strings that limits the set of disks that an ASM instance discovers.

- Uses the default value of NULL
  - This means that ASM searches a default path looking for disks that it has read and write access to.
    - The default search path is platform-specific.
    - The default search path includes Oracle ASMLib disks.
- Can use * and ? as wildcards and regular expression syntax
- Can be set dynamically by using ALTER SYSTEM
  - A change is rejected if the proposed value cannot be used to discover *all* the disks that belong to the currently mounted disk groups.

The ASM_DISKSTRING initialization parameter specifies a comma-delimited list of strings that limits the set of disks that an ASM instance discovers. Only disks that match one of the strings are discovered. The same disk cannot be discovered twice.

The format of the discovery string depends on the Oracle ASM library and the operating system that are in use. Pattern matching is supported. Refer to your operating system–specific installation guide for information about default pattern matching.

For example, on a Linux server that does not use ASMLib, to limit the discovery process to include only those disks that are in the /dev/rdsk/mydisks directory, set the ASM_DISKSTRING initialization parameter to:

```
/dev/rdsk/mydisks/*
```

The asterisk is required. To limit the discovery process to include only those disks that have a name that ends in disk3 or disk4, you could set ASM_DISKSTRING as follows on a Linux system:

```
ASM_DISKSTRING = '/dev/rdsk/*disk3', '/dev/rdsk/*disk4'
```

The ? character, when used as the first character of a path, expands to the Oracle home directory. Depending on the operating system, when you use the ? character elsewhere in the path, it is a wildcard for one character.

The default value of the ASM_DISKSTRING parameter is a NULL string. A NULL value causes Oracle ASM to search a default path for all the disks in the system to which the Oracle ASM instance has read and write access. The default search path is platform-specific. Refer to your operating system–specific installation guide for more information about the default search path.

Oracle ASM cannot use a disk unless all the Oracle ASM instances in the cluster can discover the disk through one of their own discovery strings. The names do not have to be the same on every node, but all the disks must be discoverable by all the nodes in the cluster. This may require dynamically changing the initialization parameter to enable adding new storage.

ASM_POWER_LIMIT

The ASM_POWER_LIMIT initialization parameter specifies the default power for disk rebalancing in the ASM instance.

- The default is 1.
  - This means that ASM conducts rebalancing operations by using minimal system resources.
- The allowable range is 0 to 1024.
  - 0 disables rebalancing operations.
  - Lower values use fewer system resources but result in slower rebalancing operations.
  - Higher values use more system resources to achieve faster rebalancing operations.
- It can be set dynamically by using ALTER SYSTEM or ALTER SESSION.

ORACLE

The ASM_POWER_LIMIT initialization parameter specifies the default power for disk rebalancing. The default value is 1 and the range of allowable values is 0 to 1024. A value of 0 disables rebalancing. Higher numeric values enable the rebalancing operation to complete more quickly, but at the cost of greater system resource usage and I/O overhead. Each instance can have a different value for ASM_POWER_LIMIT.

You can also specify the power of the rebalancing operation in a disk group with the POWER clause of the SQL ALTER DISKGROUP .. REBALANCE statement. The range of allowable values for the POWER clause is the same as that for the ASM_POWER_LIMIT initialization parameter. For Oracle Database 11*g Release 2 (11.2.0.1)* or earlier, the range of values for ASM_POWER_LIMIT is 0 to 11. If the value of the POWER clause is specified to be greater than 11 for a disk group with ASM compatibility set to less than 11.2.0.2, a warning is displayed and a POWER value equal to 11 is used for rebalancing.

Specifying the power of the rebalancing operation in a disk group affects only the rebalance operations, not the new allocations to a disk group.

INSTANCE_TYPE

INSTANCE_TYPE specifies whether the instance is a database instance or an ASM instance.

- Set INSTANCE_TYPE = ASM for an ASM instance.
- This is the only mandatory parameter setting for ASM.

The INSTANCE_TYPE initialization parameter must be set to ASM for an ASM instance. This is a required parameter and cannot be modified. The following is an example of the INSTANCE_TYPE parameter in the initialization file:

```
INSTANCE_TYPE = ASM
```

MEMORY_TARGET

MEMORY_TARGET specifies the total memory used by an ASM instance.

- Oracle strongly recommends that you use automatic memory management for ASM.
- All other memory-related instance parameters are automatically adjusted based on MEMORY_TARGET.
- The default value that is used for MEMORY_TARGET is acceptable for most environments.
- The minimum value for MEMORY_TARGET is 1 GB.
- It can be increased dynamically by using ALTER SYSTEM.

ORACLE

Oracle strongly recommends that you use automatic memory management (AMM) for ASM. AMM automatically manages the memory-related parameters for ASM instances with the MEMORY_TARGET parameter. AMM is enabled by default on ASM instances, even when the MEMORY_TARGET parameter is not explicitly set. The default value that is used for MEMORY_TARGET is acceptable for most environments. This is the only parameter that you need to set for complete ASM memory management.

You can also increase MEMORY_TARGET dynamically, up to the value of the MEMORY_MAX_TARGET parameter, just as you can for a database instance.

For Linux environments, AMM will not work if /dev/shm is not available or is undersized. You can adjust this by adding a size option to the entry for /dev/shm in /etc/fstab. For more details, see the man page for the mount command.

For recommended settings of memory initialization parameters in an Oracle Exadata environment, refer to the Oracle Exadata documentation.

Adjusting ASM Instance Parameters in SPFILEs

- The server parameter file (SPFILE) is a binary file that cannot be edited by using a text editor.
- Use the `ALTER SYSTEM` SQL command to adjust the ASM instance parameter settings in an SPFILE.

```
SQL> ALTER SYSTEM SET ASM_DISKSTRING='ORCL:*'
  2>   SID='*' SCOPE=SPFILE;
```

- In a clustered ASM environment, SPFILEs should reside in ASM or a cluster file system.

You can use a server parameter file (SPFILE) as the ASM instance parameter file. The server parameter file is a binary file that cannot be edited by using a text editor.

You can use the `ALTER SYSTEM` SQL command to adjust the ASM instance parameter settings in an SPFILE.

For example, to adjust your SPFILE so that your ASM environment discovers only Oracle ASMLib disks, you could execute:

```
ALTER SYSTEM SET ASM_DISKSTRING='ORCL:*' SID='*' SCOPE=SPFILE;
```

Using `SCOPE=SPFILE` changes the parameter setting stored in the SPFILE only. It does not alter the current parameter setting for the running ASM instance. If the parameter can be altered dynamically, you can use `SCOPE=MEMORY` to adjust a parameter for a running instance or use `SCOPE=BOTH` (or omit the `SCOPE` clause) to dynamically adjust a parameter and save the change in the SPFILE.

You can add an optional `SID` clause to specify that the setting is instance-specific or use the default `SID='*'` to explicitly state that the parameter applies to all instances. For example, to adjust your SPFILE so that the `+ASM1` instance uses a specific power-limit setting, execute:

```
ALTER SYSTEM SET ASM_POWER_LIMIT=5 SCOPE=SPFILE SID='+ASM1';
```

If you use an SPFILE in a clustered ASM environment, you should place the SPFILE in ASM, a shared Network-Attached Files (NFS) system, or on a cluster file system.

Starting and Stopping the Node Listener

- Using the `lsnrctl` utility:

```
$ lsnrctl start listener
LSNRCTL for Linux: Version 12.2.0.1.0-Production on 31-JUL-2013 16:45:35
Copyright (c) 1991, 2013, Oracle.  All rights reserved.
Starting /u01/app/12.2.0/grid/bin/tnslsnr: please wait......
     Intermediate output removed ...
The command completed successfully
$
```

- Using the `srvctl` utility (preferred):

```
$ srvctl start listener -n host01
$
```


A standard clustered ASM installation configures an Oracle network listener under the Grid home directory. This listener can be manually started and stopped by using the `lsnrctl` utility that is installed as part of the ASM installation:

```
$ lsnrctl start listener
$ lsnrctl stop listener
```

You can alternatively use the Server Control utility (`srvctl`) to start and stop the ASM listener as follows:

```
$ srvctl start listener -n <node>
$ srvctl stop listener -n <node>
```

The `lsnrctl` and `srvctl` utilities exist in both the *`Grid_home`* and RDBMS home directories; which one you use depends on where the listener configuration files reside. The Grid Infrastructure installation starts a listener with the configuration files in *`Grid_home`*. By default, the database installation uses that listener. In this case, set the `ORACLE_HOME` and `PATH` environment variables to use *`Grid_home`*, and then run the utilities.

If you create a new listener with the configuration files in the RDBMS home, set the `ORACLE_HOME` and `PATH` environment variables to use the RDBMS home, and then run the utilities.

ASM System Privileges

- An ASM instance does not have a data dictionary, so the only way to connect to ASM is by using these system privileges:

ASM Privilege	Privilege Group	Privilege
SYSASM	OSASM	Full administrative privilege
SYSDBA	OSDBA for ASM	Access to data stored on ASM Create and delete files. Grant and revoke file access.
SYSOPER	OSOPER for ASM	Limited privileges to start and stop the ASM instance along with a set of nondestructive ALTER DISKGROUP commands

- The SYS user on ASM is automatically created with the SYSASM privilege.

An ASM instance does not have a data dictionary, so the only way to connect to an ASM instance is by using one of three system privileges: SYSASM, SYSDBA, or SYSOPER. The table in the slide introduces these ASM system privileges.

The SYSDBA privilege on the ASM instance grants access to data stored on ASM. To use SQL*Plus commands to manage ASM components associated with the database, connect as SYSDBA to the database instance rather than the ASM instance. Users connected as SYSDBA can create and delete files, aliases, directories, and templates; examine various ASM instance views; operate on files that were created by this user; access files to which another user has explicitly granted access; and grant ASM file access control to other users. Users connected with the SYSDBA privilege cannot create or resize a disk group.

Note: By default, ASMCMD attempts to connect as SYSDBA based on the OS group.

Users who are granted the SYSOPER privilege on the ASM instance are allowed to start up, shut down, mount, dismount, and check disk groups (but not repair). Other operations, such as CREATE DISKGROUP and ADD/DROP/RESIZE DISK (or ASMCMD MKDG/CHDG), require the SYSASM privilege and are not allowed with the SYSOPER privilege.

During installation, the ASMSNMP user with SYSDBA privileges is created for monitoring the Oracle ASM instance.

Using a Single OS Group for Oracle ASM Users

Role/Software Owner	User	Groups/Privileges
Oracle ASM administrator/Oracle Grid Infrastructure home	`oracle`	`dba/ SYSASM, SYSDBA, SYSOPER`
Database administrator 1/Database home 1	`oracle`	`dba/ SYSASM, SYSDBA, SYSOPER`
Database administrator 2/Database home 2	`oracle`	`dba/ SYSASM, SYSDBA, SYSOPER`
Operating system disk device owner	`oracle`	`dba`

Oracle Grid Infrastructure and Oracle Database may be installed with a single OS group for all ASM users.

For example, the `oracle` user could install both the Grid Infrastructure and the Database software, and thus become the owner of both sets of software. The recommended groups for the `oracle` user in this case are the primary group `oinstall` and the secondary group `dba`.

The disk devices are owned by the `oracle` user. The `oracle` OS user may connect to the databases as `SYSDBA`, and the ASM instance as `SYSASM`. There is no separation of duties. The same user can perform all management functions on the ASM instance and both databases.

Using Separate OS Groups for Oracle ASM users

Role/Software Owner	User	Groups/Privileges

Oracle ASM administrator / Oracle Grid Infrastructure home	`grid`	`asmadmin(OSASM)/SYSASM`
		`asmdba(OSDBA for ASM)/SYSDBA`
		`asmoper(OSOPER for ASM)/SYSOPER`
		`dba1, dba2, ... (OSDBA for the databases when in an Oracle Restart configuration)`

Database administrator 1 / Database home 1	`oracle1`	`asmdba(OSDBA for ASM)/SYSDBA`
		`oper1(OSOPER for database 1)/SYSOPER`
		`dba1(OSDBA for database 1)/SYSDBA`

Database administrator 2 / Database home 2	`oracle2`	`asmdba(OSDBA for ASM)/SYSDBA`
		`oper2(OSOPER for database 2)/SYSOPER`
		`dba2(OSDBA for database 2)/SYSDBA`

Operating system disk device owner	`grid`	`asmadmin (OSASM)`

When you implement separate administrator privileges, choose an OSDBA group for the ASM instance that is different from the groups that you select for the database instance. For example, `dba1` and `dba2` are the privileged groups for database1 and database2, respectively; `asmadmin` is the privileged group for ASM. Notice that both database owners are members of the `asmdba` group. This allows both of them privileges to create, access, and manage ASM data files, but not ASM disk groups.

In this example, it is assumed that the storage administrators and database administrators are different people belonging to different groups. The database administrators of database1 are different from the administrators of database2.

Authentication for Accessing ASM Instances

There are three modes of connecting to ASM instances:

- Local connection by using operating system authentication

```
$ sqlplus / AS SYSASM
```

```
SQL> CONNECT / AS SYSOPER
```

- Local connection by using password file authentication

```
$ sqlplus fred/xyzabc AS SYSASM
```

```
SQL> CONNECT bill/abc123 AS SYSASM
```

- Remote connection by using Oracle Net Services and password authentication

```
$ sqlplus bill/abc123@asm1 AS SYSASM
```

```
SQL> CONNECT fred/xyzabc@asm2 AS SYSDBA
```

ORACLE

There are three modes of connecting to ASM instances:

- **Local connection by using operating system authentication:** Operating system users that are members of the `OSASM`, `OSDBA` for ASM, or `OSOPER` for ASM groups can connect to ASM without providing any additional credentials. For example, an operating system user who is a member of the `OSASM` group can connect to ASM with full administrative privileges by using:
  CONNECT / AS SYSASM
  Note: A local connection by using `AS SYSDBA` always uses OS authentication even when using the username/password syntax.
- **Local connection by using password file authentication:** The following example shows a local connection by using password file authentication:
  CONNECT sys/<sys_password> AS SYSASM
- **Remote connection by way of Oracle Net Services by using password authentication:** Password-based authentication is also supported remotely by using Oracle Net Services. The following example shows a remote connection by way of Oracle Net Services by using password authentication.
  CONNECT sys/<sys_password>@<net_services_alias> AS SYSASM

ASMCMD and Authentication

- Before running ASMCMD, log in to the host containing the Oracle ASM instance that you plan to administer.
  - You must log in as a user that has SYSASM or SYSDBA privileges through operating system authentication.
  - The SYSASM privilege is the required connection to administer the Oracle ASM instance.
- If the ASM instance is not running, ASMCMD runs only those commands that do not require an ASM instance.
  - These include startup, shutdown, lsdsk, help, and exit.
- You can connect as SYSDBA by running ASMCMD that is located in the bin directory of the Oracle Database home.
  - Include the --privilege option to connect as SYSDBA.

```
$ asmcmd --privilege sysdba
```

ORACLE

Before running ASMCMD, log in to the host that contains the Oracle ASM instance that you plan to administer. You must log in as a user that has SYSASM or SYSDBA privileges through operating system authentication. The SYSASM privilege is the required connection to administer the Oracle ASM instance.

To connect to the Oracle ASM instance, run ASMCMD that is located in the bin subdirectory of the Oracle Grid Infrastructure home (Oracle ASM home).

Connect as SYSASM, the default connection, to administer an Oracle ASM instance.

Ensure that the ORACLE_HOME and ORACLE_SID environment variables refer to the Oracle ASM instance. Ensure that the bin subdirectory of your Oracle Grid Infrastructure home is in your PATH environment variable. To use most of the ASMCMD commands, ensure that the Oracle ASM instance is started and the Oracle ASM disk groups are mounted.

If the Oracle ASM instance is not running, ASMCMD runs only those commands that do not require an Oracle ASM instance. The commands include startup, shutdown, lsdsk, help, and exit.

You can connect as SYSDBA by running ASMCMD that is located in the bin directory of the Oracle Database home. Ensure that the ORACLE_HOME and ORACLE_SID environment variables refer to the database instance.

You must include the --privilege option to connect as SYSDBA. For example:

```
$ asmcmd --privilege sysdba
```

When administering disk groups, Oracle recommends that you run ASMCMD from the database home of the database instance that is the owner of the files in the disk group.

Password-Based Authentication for ASM

- Password-based authentication:
  - Uses a password file
  - Can work both locally and remotely
    - `REMOTE_LOGIN_PASSWORDFILE` must be set to a value other than `NONE` to enable remote password-based authentication.
- A password file is created initially:
  - By Oracle Universal Installer when installing ASM
  - Manually with the `orapwd` utility
  - Containing only the `SYS` and `ASMSNMP` users
- Users can be added to the password file by using:
  - The SQL*Plus `GRANT` command
  - The ASMCMD `orapwusr` command

For ASM, password-based authentication uses a password file. Password-based authentication works both locally and remotely. Password-based authentication is enabled by default when the ASM instance is created by Oracle Universal Installer. You can create a password file for ASM manually by using the `orapwd` utility.

To enable remote password-based authentication, the password file must exist, and the initialization parameter `REMOTE_LOGIN_PASSWORDFILE` must have a value other than `NONE`. The default value is `EXCLUSIVE` for ASM instances.

To add other users to the password file, you can use the SQL `CREATE USER` and `GRANT` commands. For example:

```
REM create a new user, then grant the SYSOPER privilege
SQL> CREATE USER new_user IDENTIFIED BY new_user_passwd;
SQL> GRANT SYSOPER TO new_user;
```

With ASMCMD, add a user to the password file with the `SYSASM` privilege:

```
asmcmd orapwusr --add --privilege sysasm new_user
```

Managing the ASM Password File

For the ASM instance, the password file:

- Can be created by a user that owns the ASM software
- Holds roles assigned to users
- Is required for Oracle Enterprise Manager to connect to ASM remotely
- Can be viewed from
  - SQL*Plus `SELECT * FROM V$PWFILE_USERS`
  - ASMCMD `lspwusr`

ORACLE

A password file is required to enable Oracle Enterprise Manager to connect to ASM.

You can list the users in the password file and their assigned privileges with the SQL*Plus command:

```
SELECT * FROM V$PWFILE_USERS;
```

or from ASMCMD with:

```
lspwusr
```

To revoke a privilege, use the SQL*Plus command:

```
REVOKE SYSASM FROM user
```

In ASMCMD, you can change the privilege with:

```
asmcmd orapwuser --modify --privilege sysasm user
```

Note: Whatever privilege is named replaces any other privilege that was previously granted.

In ASMCMD, you can remove a user from the password file with:

```
asmcmd orapwusr --delete user
```

Note: ASMCMD users cannot connect through the password file. Their privileges are determined by the OS group membership.

Managing a Shared Password File in a Disk Group

- A password file for Oracle Database or Oracle ASM can reside on a designated Oracle ASM disk group.
- Having the password files reside on a single location that is accessible across the cluster reduces:
  - Maintenance costs
  - Situations where passwords become out of sync
- The `COMPATIBLE.ASM` disk group attribute must be at least 12.1 for the disk group where the password is located.
- Specify the disk group location and database unique name when using `orapwd` to create a database password file.

```
$ orapwd file='+data/ORCL/orapwdb' dbuniquename='orcl'
```

- For an ASM password file, use the `asm` switch with `orapwd`.

```
$ orapwd file='+data/ASM/orapwasm' asm=y
```

ORACLE

An individual password file for Oracle Database or Oracle ASM can reside on a designated Oracle ASM disk group. Having the password files reside on a single location that is accessible across the cluster reduces maintenance costs and situations where passwords become out of sync.

You can use a password file that is located on a disk group for authentication only if the Oracle ASM instance is running and the designated disk group is mounted. Otherwise, operating-system authentication must be used to bootstrap the startup of the Oracle ASM instance and stack.

The `COMPATIBLE.ASM` disk group attribute must be set to at least 12.1 for the disk group where the password is to be located. The `SYSASM` privilege is required to manage the Oracle ASM password file. The `SYSDBA` privilege on Oracle ASM is required to manage the database password file.

The shared password file in a disk group is managed by `asmcmd` commands, the `orapwd` tool, and `srvctl` commands. `orapwd` supports the creation of password files in an Oracle ASM disk group. All other password file manipulation is performed with `asmcmd` or `srvctl` commands.

You must specify the disk group location and database unique name when using `orapwd` to create a database password file in a disk group.

For example:

```
$ orapwd file='+data/ORCL/orapwdb' dbuniquename='orcl'
```

The `asm` switch specifies that `orapwd` create an Oracle ASM password file rather than a database password file. For example:

```
$ orapwd file='+data/ASM/orapwasm' asm=y
```

You can create a new password file in a disk group by using a password file from a previous release. For example:

```
$ orapwd input_file='/oraclegrid/dbs/orapwasm' file='+data/ASM/orapwasm' asm=y
```

The `srvctl` commands include updates to manage a password file in a disk group, such as the following for updating and displaying the location of the password file:

```
$ srvctl modify asm -pwfile location
```

```
$ srvctl modify database -db dbname -pwfile location
$ srvctl config asm
ASM home: /u01/app/12.2.0/grid
Password file: +DATA/orapwASM
ASM listener: LISTENER
ASM instance count: 3
Cluster ASM listener: ASMNET1LSNR_ASM
```

ASM Dynamic Performance Views

The ASM instance hosts memory-based metadata tables that are presented as dynamic performance views.

- The instance is accessed by ASM utilities to retrieve metadata-only information by using SQL language.
- It contains many dedicated ASM-related views such as:

```
V$ASM_ALIAS        V$ASM_ATTRIBUTE       V$ASM_CLIENT
V$ASM_DISK         V$ASM_DISK_IOSTAT     V$ASM_DISK_STAT
V$ASM_DISKGROUP    V$ASM_DISKGROUP_STAT  V$ASM_FILE
V$ASM_OPERATION    V$ASM_TEMPLATE        V$ASM_USER
V$ASM_FILESYSTEM V$ASM_USERGROUP  V$ASM_USERGROUP_MEMBER
```

The `V$ASM_*` views exist in both ASM and database instances. The rows returned will vary.

ORACLE

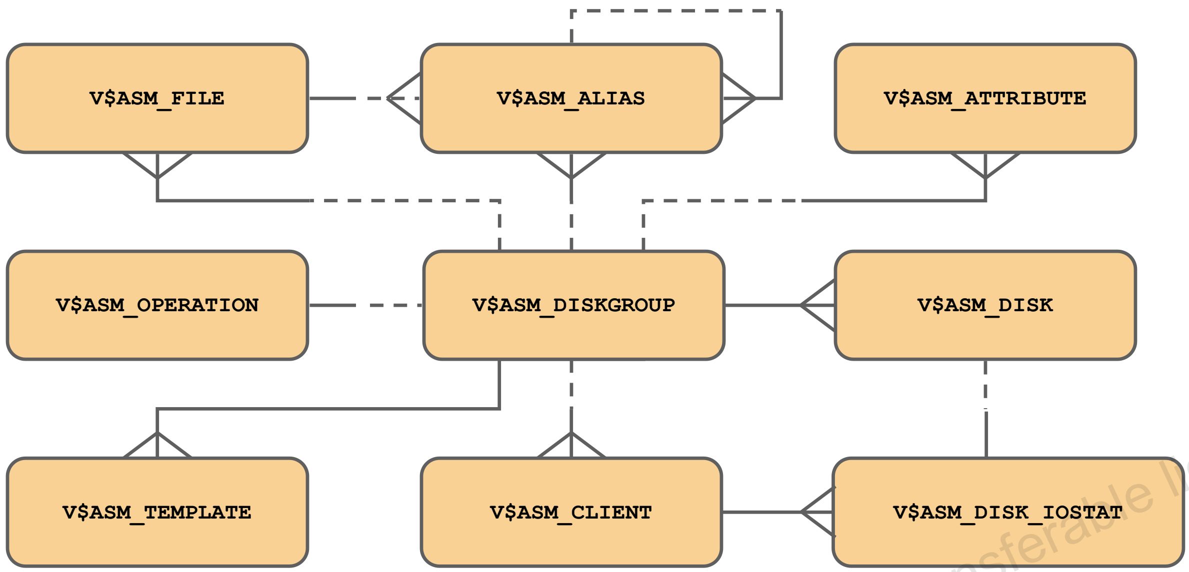
An ASM instance does not have a data dictionary. Instead, information about its configuration and operation is available by querying a set of dynamic performance views. The ASM dynamic performance views have names with the prefix `V$ASM`. They can be queried by using SQL in the same manner as any database dynamic performance view. The slide lists the most common dynamic performance views that contain ASM-related metadata.

The `V$ASM` views exist inside both ASM instances and database instances. If you query a `V$ASM` view while connected to an ASM instance, information relating to the ASM instance is returned. If you query a `V$ASM` view while connected to a database instance, some views return no information, whereas others return a subset of information relating to the use of ASM in that database instance.

In a clustered environment, in addition to the `V$ASM` views, there is a corresponding set of `GV$ASM` views. The `GV$ASM` views provide a consolidated view of information across all the currently running clustered ASM instances. The structure of each `GV$ASM` view is the same as for its corresponding `V$ASM` view except for an additional column, `INST_ID`, which reports the instance identifier that relates to each record.

Note: ASM dynamic performance views are referenced later in this course in the context of performing various administrative tasks. For a complete definition of all the columns in each view, refer to *Oracle Database Reference 12c Release 1* in the Oracle Database Documentation Library.

ASM Dynamic Performance Views Diagram



The diagram in the slide shows the relationships between the ASM dynamic performance views. It is noteworthy that all the views in some way relate back to `V$ASM_DISKGROUP` through the `GROUP_NUMBER` column.

The relationships shown in the diagram may be expressed by using the following statements:

- A disk group listed in `V$ASM_DISKGROUP` may contain numerous files listed in `V$ASM_FILE`.
- A disk group listed in `V$ASM_DISKGROUP` may contain numerous aliases listed in `V$ASM_ALIAS`.
- A file record listed in `V$ASM_FILE` will have a file name and may have an alias listed in `V$ASM_ALIAS`.
- A file name listed in `V$ASM_ALIAS` will relate to a file record contained in `V$ASM_FILE`.
- A file alias listed in `V$ASM_ALIAS` will relate to a file record contained in `V$ASM_FILE`.
- A directory alias listed in `V$ASM_ALIAS` will not have a corresponding record contained in `V$ASM_FILE`.
- A directory alias listed in `V$ASM_ALIAS` may be the parent of numerous file names and file aliases listed in `V$ASM_ALIAS`.
- A disk group listed in `V$ASM_DISKGROUP` will have attributes associated with it listed in `V$ASM_ATTRIBUTE` if the disk group attribute `COMPATIBLE.ASM` is set to 11.1 or higher.
- A disk group listed in `V$ASM_DISKGROUP` may be the subject of a long-running operation listed in `V$ASM_OPERATION`.
- A disk group listed in `V$ASM_DISKGROUP` will contain numerous templates listed in `V$ASM_TEMPLATE`.

- A disk group listed in V$ASM_DISKGROUP might be currently used by numerous database instances listed in V$ASM_CLIENT.
- A disk group listed in V$ASM_DISKGROUP will contain numerous disks listed in V$ASM_DISK.
- A database client listed in V$ASM_CLIENT will have numerous performance records listed in V$ASM_DISK_IOSTAT, each of which corresponds to a disk listed in V$ASM_DISK.

Note: The V$ASM_DISKGROUP_STAT and V$ASM_DISK_STAT views are not shown here. These views mirror V$ASM_DISKGROUP and V$ASM_DISK, respectively. These views are described in more detail in *Oracle Database Reference 12c Release 1*.

Quiz

Which two are recommended configurations for ASM instance initialization?

a. Store the SPFILE on a shared raw device.
b. Use a server parameter file (SPFILE).
c. Store the SPFILE on separate disks.
d. Store the SPFILE in an ASM disk group.
e. Use a text initialization parameter file (PFILE).
f. Use a PFILE that references an SPFILE.

Answer: b, d

Quiz

Automatic memory management is enabled by default on ASM instances.

a. True

b. False

Answer: a

Oracle strongly recommends that you use automatic memory management for ASM.

The `MEMORY_TARGET` value is set during installation and may be adjusted later.

Summary

In this lesson, you should have learned how to:

- Explain and apply ASM initialization parameters
- Manage ASM instances and associated processes
- Monitor ASM

Practice 2 Overview: Administering ASM Instances

This practice covers the following topics:

- Adjusting initialization parameters
- Stopping and starting instances
- Monitoring the status of instances

ORACLE

3

Flex ASM

Objectives

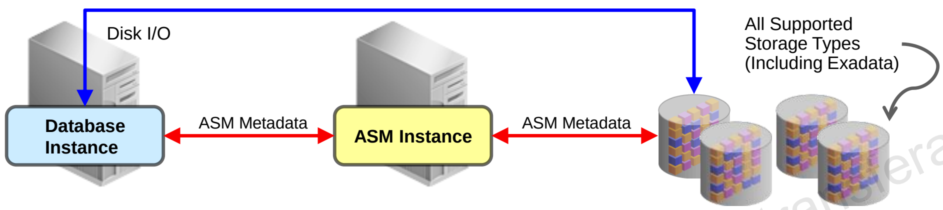
After completing this lesson, you should be able to:

- Describe the architecture and components of Flex ASM
- Configure Flex ASM
- Administer Flex ASM

ORACLE

Flex ASM: Overview

- In earlier versions, ASM clients can access ASM only by using an ASM instance on the same host.
  - Resources are consumed by ASM on every database server.
  - If an ASM instance fails, its clients must fail.
- With Flex ASM, ASM clients can use a network connection to access ASM.
  - Resources are saved because ASM is not required on every database server.
  - If an ASM instance fails, its clients can connect to another instance.



Before Oracle Database 12*c*, an ASM client (database instance or ASM Cluster File System) can connect only to an ASM instance that is running on the same host. This requires every database server to dedicate system resources to ASM, which increases the overall system resource requirement to run Oracle Database in conjunction with ASM. This tightly coupled model also has availability concerns because if an ASM instance fails, all ASM clients on that host must also fail.

Oracle Database 12*c* introduces Flex ASM. Flex ASM allows ASM clients to connect to ASM over a network. By relaxing the hard dependency between ASM and its clients, the previous architectural limitations are overcome. With Flex ASM, a smaller pool of ASM instances can be used to serve a large pool of database servers. If an ASM instance fails, its clients can reconnect to another ASM instance.

Flex ASM and Flex Clusters

- Flex Clusters require Flex ASM.
  - Standard ASM is not supported on a Flex Cluster.
- Flex ASM does not require a Flex Cluster.
  - Flex ASM can run on a standard cluster, servicing clients across the cluster.
  - Flex ASM can run on the Hub Nodes of a Flex Cluster, servicing clients across the Hub Nodes of the Flex Cluster.
- The benefits of Flex ASM apply regardless of cluster type:
  - Smaller ASM resource footprint
  - Protection from ASM failure

ORACLE

Standard ASM is not supported on a Flex Cluster. Therefore, Flex Clusters require Flex ASM. However, Flex ASM does not require a Flex Cluster. Flex ASM can be configured on either a standard cluster or a Flex Cluster.

When Flex ASM runs on a standard cluster, ASM services can run on a subset of cluster nodes servicing clients across the cluster. When Flex ASM runs on a Flex Cluster, ASM services can run on a subset of Hub Nodes servicing clients across all the Hub Nodes in the Flex Cluster.

The fundamental benefits of Flex ASM apply regardless of the type of cluster being used. That is:

- The overall resource footprint is smaller because a smaller pool of ASM instances can be used to serve a larger pool of database servers
- Higher availability can be achieved because if an ASM instance fails, its clients can reconnect to another ASM instance

Oracle ASM Flex Configurations

An Oracle Flex ASM instance can operate in several configurations:

- Local Oracle ASM clients with direct access to Oracle ASM disks
- Oracle Flex ASM clients with direct access to ASM disks
- Oracle ACFS access through the Oracle ASM proxy instance
- Network-based connectivity to ASM disk groups with ASM IOServer (IOS)

ORACLE

Local Oracle ASM clients with direct access to Oracle ASM disks

- Flex ASM can be configured with local mode if the number of hub nodes is less than, equal to cardinality, or cardinality is set to ALL. In this configuration, the database instances are on the same hub node as the Oracle ASM instance and are referred to as local Oracle ASM client instances. Oracle ASM metadata moves between Oracle ASM and the database instances. This client has direct I/O access to Oracle ASM disks.
- Oracle database 12*c* instances are allowed to connect to the remote ASM instances in case of server failure, however pre-12*c* database instances can only connect to a local ASM instance.

Oracle Flex ASM clients with direct access to Oracle ASM disks

- In this configuration, the database instances are on different host computers than the nearby Oracle ASM instance, and are referred to as Oracle ASM client instances.
- The databases are in the same Oracle ASM cluster as the Oracle ASM instance and the database instances are located on a Hub node. Oracle ASM metadata moves between Oracle ASM and the database instance. This client has direct I/O access to Oracle ASM disks.
- Depending on the distribution of database instances and Oracle ASM instances, a database client may access Oracle ASM locally on the same node or remotely over the Oracle ASM network. This mode of operation is used by database clients on Hub nodes in the Oracle ASM cluster. Direct access mode is also the only Oracle Flex ASM configuration supported by Oracle ASM cluster file system.

Oracle ACFS access through the Oracle ASM proxy instance

- An Oracle ASM proxy instance is an Oracle instance running on a Hub node with a direct Oracle ASM client. An Oracle ASM proxy instance provides support for Oracle Automatic Storage Management Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM).

Network-based connectivity to Oracle ASM disk groups with Oracle IOServer (IOS)

- An Oracle IOServer instance provides Oracle ASM file access for Oracle Database instances on nodes of Oracle member clusters that do not have connectivity to Oracle ASM managed disks.

ASM Instance Changes

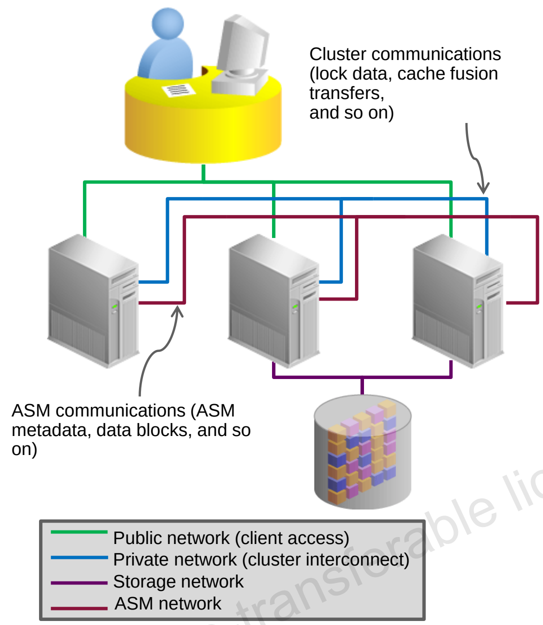
- ASM instances are no longer required to run on every node in a cluster.
- Administrators specify the cardinality for ASM.
  - Cardinality sets the maximum number of ASM instances in a given cluster.
  - The default is 3.
- All disk groups are mounted by all ASM instances.

Flex ASM relaxes the requirement for an ASM instance to exist on every node in the cluster. Instead, the administrator specifies the cardinality for ASM. This number specifies the number of ASM instances that should be made available in the ASM cluster. The default cardinality setting for ASM instances is three.

Because an ASM instance can now service any database instance across the cluster, all disk groups are typically mounted on all ASM instances. As is the case in earlier versions, the ASM instance that is running on a node has its `ORACLE_SID` set to `+ASM<node number>`.

ASM Network

- In earlier versions, a CSS cluster requires:
  - A public network for client application access
  - One or more private networks for inter-node communication within the cluster
- Flex ASM adds the ASM network, which is used for communication between ASM and ASM clients.



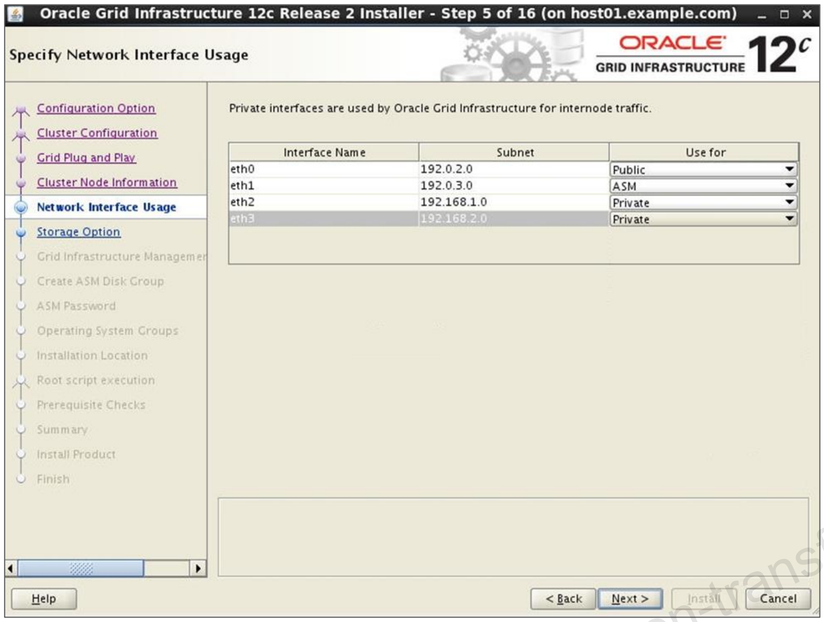
In earlier versions, a CSS cluster requires access to a public network and one or more private networks. Clients outside the cluster use the public network to connect to the servers in the cluster. The private networks are predominantly used for inter-node communication within the cluster. Sometimes, the private network also serves as the storage network. This is the case inside Oracle Exadata Database Machine.

Flex ASM introduces a new type of network called the ASM network. The ASM network is used for all communication between ASM and its clients. There can be more than one ASM network in a customer environment. ASM provides its services on all the ASM networks. This requires all ASM networks to be accessible on all the nodes that host the ASM instances.

All ASM clients running within the ASM cluster can use any of the ASM networks to communicate with ASM.

It is possible to configure a network as both a private and an ASM network. That is, a single network can perform both functions.

Configuring ASM Network: Example

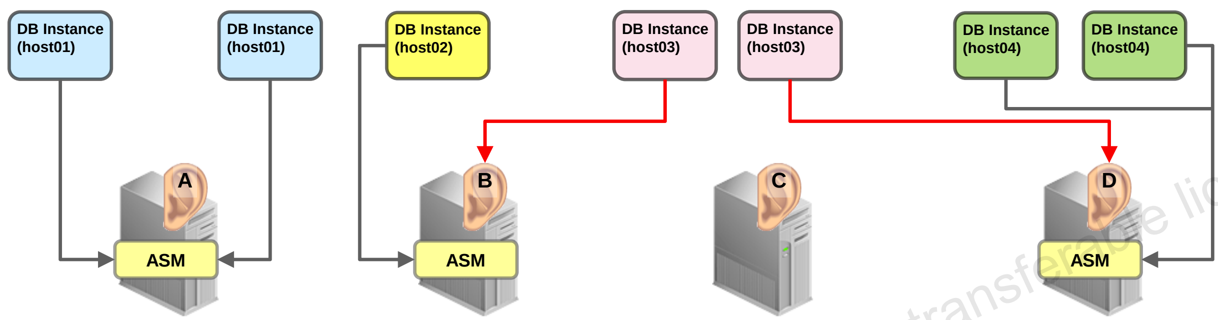


On the Specify Network Interface Usage page, you can select the network interfaces on your cluster nodes to use for internode communication. You must choose one interface for the public network and one for the ASM and the private network.

Ensure that the network interfaces that you choose for the interconnect have enough bandwidth to support the cluster and RAC-related network traffic. To configure the interfaces, click the drop-down list to the right of the interface under the “Use for” column, and select proper usage for each network interface. In the example in the slide, there are four interfaces: eth0, eth1, eth2, and eth3. The eth0 interface is the hosts’ primary network interface and should be marked Public. The eth1 interface is configured for ASM and eth2 and eth3 interfaces are configured for private network. If you have other adapters dedicated to a storage network, they should be marked Do Not Use. When you finish, click the Next button to continue.

ASM Listeners

- All local clients connect to the ASM instance bypassing the listener process
- All remote clients connect to the ASM instance using `+ASM` service, registered with the ASM listener processes.
- The automatic service cross-registration is performed automatically using `LISTENER_NETWORKS` ASM instance parameter and a new asm network



ORACLE

To support Flex ASM connectivity, a set of ASM listeners is configured for every ASM network. The ASM listeners are in addition to other listeners such as the SCAN listeners and the local database listeners. The diagram in the slide illustrates the arrangement of the ASM listeners.

Each hub node hosts one local ASM listener for an ASM network. Each ASM listener can service client connections over the corresponding ASM network. Up to three ASM listener addresses are registered as remote listeners using LISTENER_NETWORKS ASM instance parameter. This parameter is automatically modified by the oraagent process. Using this arrangement, clients have a highly available connection endpoint to facilitate connection to ASM instances.

While connection is initiated by using one of the registered remote listeners, all client connections are load balanced across the entire set of available ASM instances. The load-balancing mechanism is connect-time load balancing.

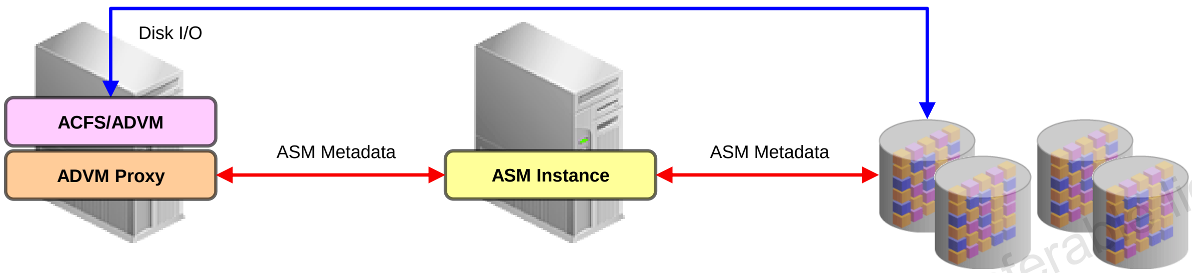
```
SQL> show parameter listener_networks

NAME                   TYPE        VALUE
---------------------- ----------- ------------------------------
listener_networks      string       ((NAME=ora.ASMNET1LSNR_ASM.lsn
                                    r)  (LOCAL_LISTENER="(DESCRIPTI
                                    ON=(ADDRESS=(PROTOCOL=TCP)(HOS
                                    T=192.0.3.101)(PORT=1526)))"))
                                     ((NAME=ora.ASMNET1LSNR_ASM.l
                                    snr)(REMOTE_LISTENER="(DESCRIP
                                    TION=(ADDRESS=(PROTOCOL=TCP)(H
                                    OST=192.0.3.101)(PORT=1526)))"
                                     )), ((NAME=ora.ASMNET1LSNR_ASM
                                   .lsnr)  (REMOTE_LISTENER="(DESC
                                   RIPTION=(ADDRESS=(PROTOCOL=TCP
                                   )(HOST=192.0.3.103)(PORT=1526)
                                   ))")), ((NAME=ora.ASMNET1LSNR_
                                   ASM.lsnr)  (REMOTE_LISTENER="(D
                                   ESCRIPTION=(ADDRESS=(PROTOCOL=
                                   TCP)(HOST=192.0.3.102)(PORT=15
                                   26)))"))
```

ADVM Proxy

The ADVM Proxy is a special Oracle instance.

- It enables ADVM to connect to Flex ASM.
- It is required to run on the same node as ADVM and ACFS.
- It can be shut down when ACFS is not running.



The ADVM proxy is a special Oracle instance. Its sole purpose is to enable ASM Dynamic Volume Manager (ADVM), and through it ASM Cluster File System (ACFS), to connect to Flex ASM.

In release 12.2, ACFS, ADVM, and the ADVM proxy must reside on the same node. So by default, the ADVM proxy is configured to run on every node in a standard cluster or every Hub Node in a Flex Cluster. However, when Flex ASM is configured by using `cardinality = ALL`), the ADVM proxy (`ora.proxy_advm`) is not required because the ASM instances are up and running on all nodes to handle ADVM volumes and ACFS file systems. Administrators can shut down the ADVM proxy if ACFS is not running on the node.

The ADVM proxy instance has its `ORACLE_SID` set to `+APX<node number>`.

Managing Flex ASM Instances

Flex ASM is designed to require minimal monitoring and ongoing management.

- The primary concern is that instances are up and running.

```
$ asmcmd showclustermode
ASM cluster : Flex mode enabled

$ srvctl status asm -detail
ASM is running on host03,host02,host01
ASM is enabled.
ASM instance +ASM1 is running on node host01
Number of connected clients: 4
Client names: +APX1:+APX:cluster01 -MGMTDB:_mgmtdb:cluster01 host01.example.com:_OCR:cluster01
orcl_3:orcl:cluster01
...
```

- No Flex ASM–specific instance parameters are required.
- The default settings will effectively support most situations.

ORACLE

Flex ASM is designed to require minimal monitoring and ongoing management after initial configuration. The primary concern for administrators is that the ASM instances are up and running. This can be verified by using the `srvctl status` commands shown in the slide.

In release 12.2, no new instance parameters are specific to Flex ASM. In addition, the default parameter settings have been adjusted to suit the Flex ASM architecture, making them sufficient to effectively support most situations.

Automatic memory management is used for ASM instances. In release 12.2, the default setting for `MEMORY_TARGET` is based on various attributes of the node that hosts the instance, such as the physical memory size and the number of processor cores.

Note that the minimum default `MEMORY_TARGET` setting (`1GB`) is significantly larger than the default `MEMORY_TARGET` setting used by ASM instances in earlier versions.

Stopping, Starting, and Relocating Flex ASM Instances

- ASM Instances

```
$ srvctl status asm -detail
ASM is running on host03,host02
ASM is enabled.
$ srvctl stop asm -node host03 -f
$ srvctl start asm -node host01
$ srvctl status asm -detail
ASM is running on host01,host02
ASM is enabled.
$ srvctl relocate asm -currentnode host01 -targetnode host03
$ srvctl status asm -detail
ASM is running on host03,host02
ASM is enabled.
```

- ADVM Proxy Instances

```
$ srvctl stop asm -proxy -node host03
$ srvctl start asm -proxy -node host01
```

ORACLE

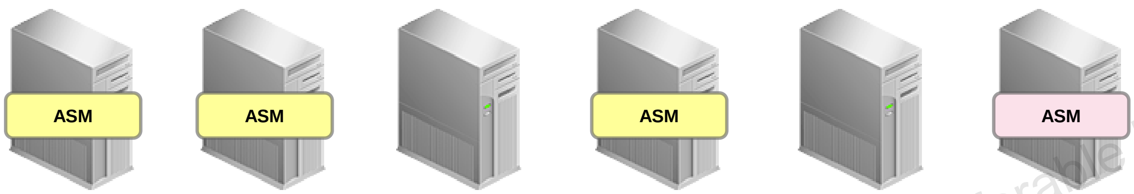
At times, it may be useful for administrators to control an individual ASM instance or ADVM Proxy instance. The slide shows examples of the `srvctl` commands to stop, start, and relocate individual Flex ASM instances.

Setting the Cardinality for Flex ASM Instances

- ASM Instances

```
$ crsctl status resource ora.asm -f | grep CARDINALITY=
CARDINALITY=3

$ srvctl modify asm -count 4
$ crsctl status resource ora.asm -f | grep CARDINALITY=
CARDINALITY=4
```



ORACLE

The slide shows examples of the commands required to manage the cardinality setting for ASM instances. To view the current cardinality setting, use the `crsctl status resource` commands shown in the slide. To set the cardinality, use the `srvctl modify` command.

You can also use the `srvctl config asm` command to view the cardinality, or number of Flex ASM instances:

```
$ srvctl config asm
ASM home: /u01/app/12.2.0/grid
Password file: +DATA/orapwASM
ASM listener: LISTENER
ASM instance count: 3
Cluster ASM listener: ASMNET1LSNR_ASM
```

Monitoring Flex ASM Connections

```
SQL> select distinct i.instance_name asm_instance_name,
  2  c.instance_name client_instance_name, c.db_name, c.status
  3  from gv$instance i, gv$asm_client c
  4  where i.inst_id=c.inst_id;

ASM_INSTANCE_NAME    CLIENT_INSTANCE_NAME DB_NAME  STATUS
-------------------- -------------------- -------- ------------
+ASM1                +APX1                +APX     CONNECTED
+ASM1                +ASM1                +ASM     CONNECTED
+ASM1                orcl_2               orcl     CONNECTED
+ASM1                orcl_5               orcl     CONNECTED
+ASM1                orcl_7               orcl     CONNECTED
+ASM2                +APX2                +APX     CONNECTED
+ASM2                +ASM2                +ASM     CONNECTED
+ASM2                orcl_1               orcl     CONNECTED
+ASM2                orcl_4               orcl     CONNECTED
+ASM3                +APX3                +APX     CONNECTED
+ASM3                +ASM3                +ASM     CONNECTED
+ASM3                orcl_3               orcl     CONNECTED
+ASM3                orcl_6               orcl     CONNECTED
+ASM3                orcl_8               orcl     CONNECTED
```

ORACLE

At times, it may be useful for administrators to know which clients are connected to each ASM instance. This knowledge may be especially useful when considering the impact of shutting down a node for scheduled maintenance or if a change in the cardinality setting for ASM instances is being considered.

To determine the database instances that are connected to a specific ASM instance, ASM administrators can connect to an ASM instance and query the `GV$ASM_CLIENT` table. The example in the slide shows the distribution of eight database instances (`orcl_1` to `orcl_8`) across three Flex ASM instances (`+ASM1`, `+ASM2`, `+ASM3`).

Relocating an ASM Client

- Clients are automatically relocated to another instance if an ASM instance fails.
  - Clients reconnect and the connection is load balanced to an available instance.
- If necessary, clients can be manually relocated by using the `ALTER SYSTEM RELOCATE CLIENT` command.
  - Command Syntax:

```
SQL> ALTER SYSTEM RELOCATE CLIENT '<instance_name>:<db_name>';
```

  - Query `GV$ASM_CLIENT` to determine *`instance_name`* and *`db_name`*.
  - Useful for manually adjusting the workload balance between instances

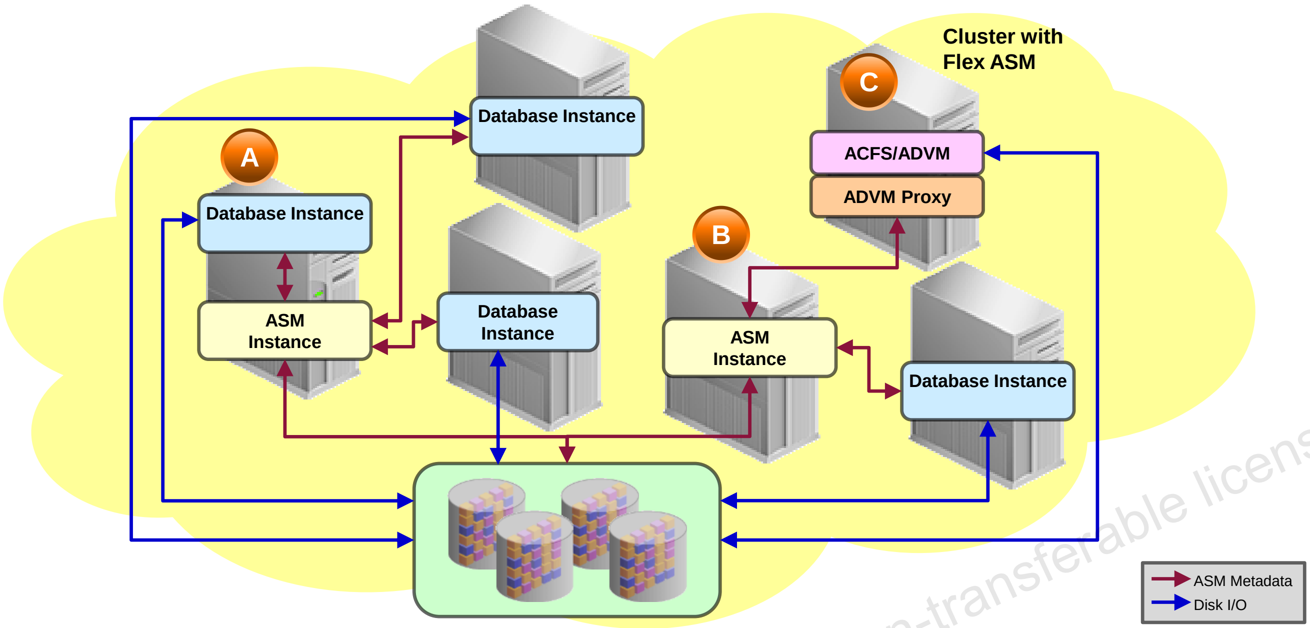
ORACLE

With Flex ASM, if an ASM instance fails, clients are automatically relocated to another instance. When a failure is detected by a client, it reconnects to an available instance. Like any other connection request, reconnection requests are subject to connection load balancing. Relocation due to failure is automatic and transparent to end users.

In addition to automatic relocation, the `ALTER SYSTEM RELOCATE CLIENT` command can be used on the node that is currently hosting the client to perform manual relocation. This command results in the server terminating the connection to the client, which forces the client to reconnect to the least loaded ASM instance. Manual relocation is useful for manually adjusting the workload balance between ASM instances when a significant imbalance is detected.

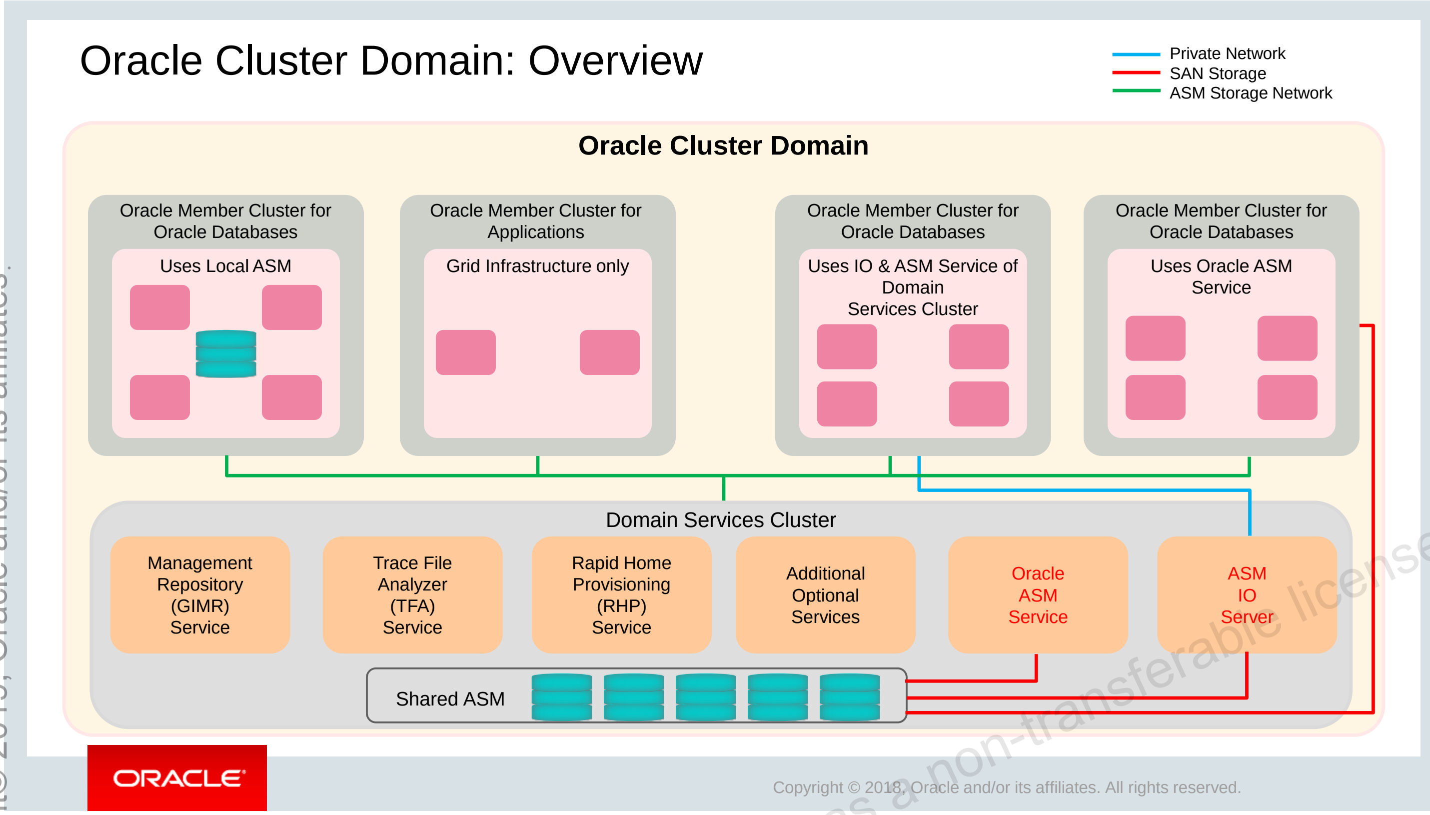
Note: If an ASM client is already connected to the least loaded ASM instance, the `ALTER SYSTEM RELOCATE CLIENT` command will cause the client to disconnect, however, it will reconnect to the same ASM instance. In this case, to force an ASM client to relocate off the ASM instance, you would need to shut down the ASM instance.

Flex ASM Deployment: Example



The diagram in the slide shows a sample Flex ASM deployment. The following notes provide additional detail and summarize some of the key points relating to Flex ASM:

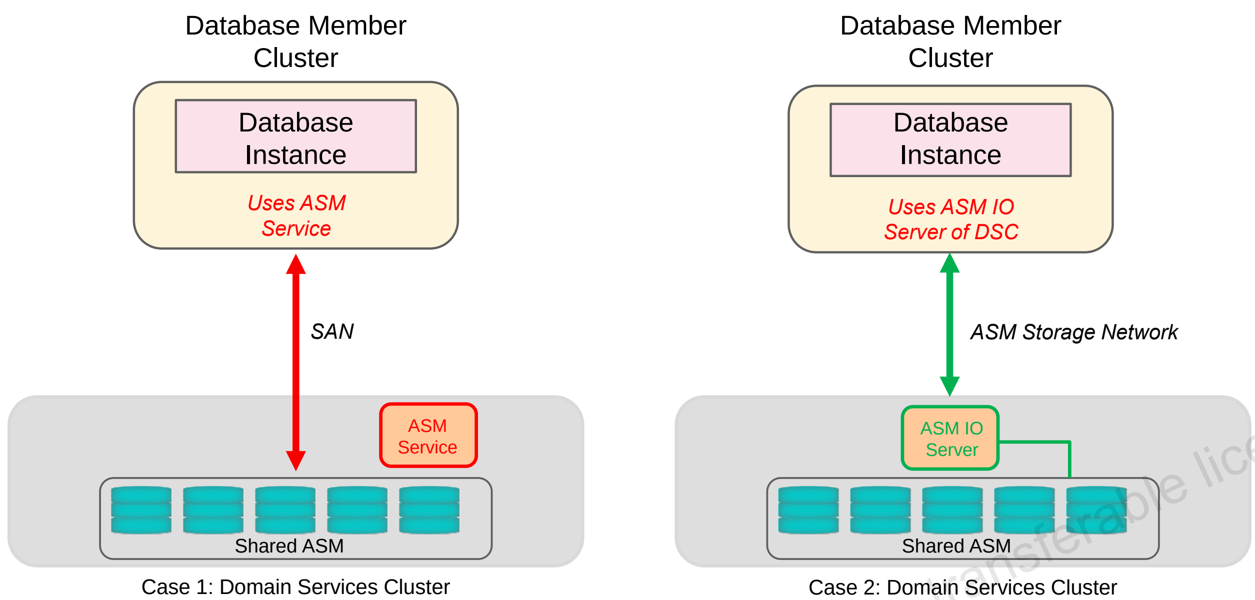
- The diagram illustrates a standard cluster that is running Flex ASM. The diagram also illustrates the Hub Nodes of a Flex Cluster.
- In release 12.2, ASM clients can run on any node in a Flex cluster.
- Flex ASM enables a smaller number of ASM instances (two in this example) to service a larger number of clients (four database instances and one ACFS in this example).
- Flex ASM enhances the availability of Oracle Database and ACFS by helping to protect against various ASM failures. If, for example, the ASM instance at node A failed, the three database instances that it supports would transparently connect to the ASM instance at node B.
- The ASM cardinality setting specifies the number of ASM instances that should be made available in the cluster. In this example, the ASM cardinality is two. The default cardinality setting for ASM instances is three.
- Depending on the distribution of clients and ASM instances, an ASM client may reside on the same node as an ASM instance (as shown on node A in the diagram), or the ASM instance may reside on a node that is separate from the ASM clients (as shown on node B in the diagram).



The diagram in the slide shows the new cluster configuration option in 12.2, which is the Oracle Cluster Domain. Simply stated, a *Cluster Domain* is collection of independent clusters, called *Member Clusters*, and a centralized *Domain Services Cluster* (DSC) providing centralized ASM and other management services. When a Cluster Domain is deployed, a DSC can eliminate the requirement for ASM instances to be running on the Member Clusters by utilizing ASM shared storage services in the DSC. However, while Member Clusters can use the ASM services in the DSC, as an alternative, Member Clusters, for storage isolation reasons, are able to host their own ASM environment.

For more information about Oracle Cluster Domain, refer to *Oracle Grid Infrastructure Installation and Upgrade Guide, 12c Release 2 (12.2)*

ASM IO Services



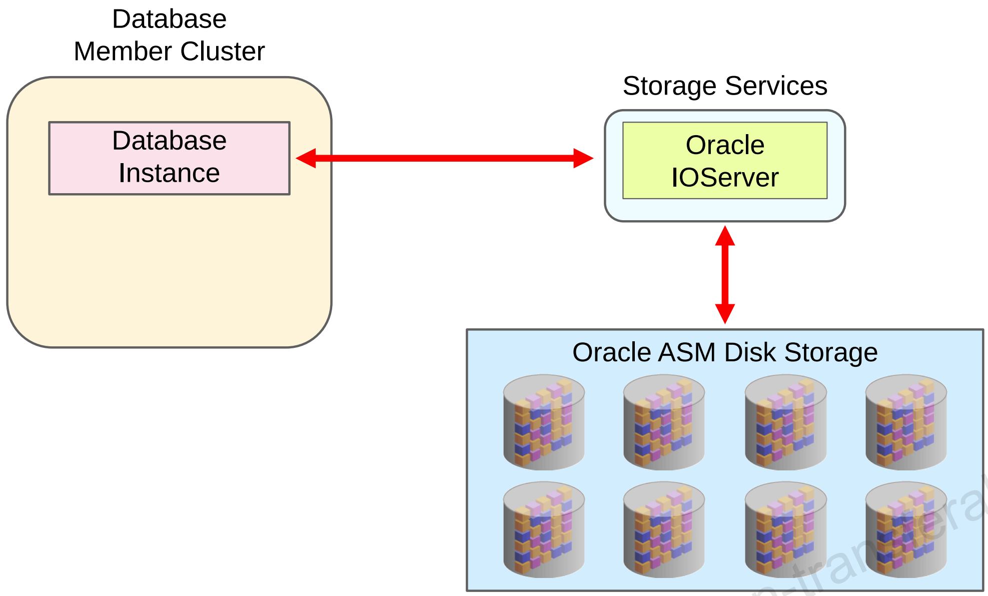
ASM Services available in Oracle Database 12*c* release 2 enables Member Clusters to access data in a centralized ASM Diskgroup environment of the Domain Services Cluster (DSC). When ASM Services in a DSC are used, there are two connectivity options by which Members Clusters can access the shared Diskgroups residing in a DSC.

The first is for the Member Cluster to have shared SAN attachment to the DSC storage as shown in case 1. The database instances in the Member Cluster rely on ASM instance(s) in the DSC for coordinating access through the SAN.

The second way in which a Member Cluster can access a DSC's Diskgroup is though *ASM IO Servers* using the *ASM Storage Network*. This is shown in case 2. With this mode, there is no physical SAN connection of storage between the Member Clusters and the DSC. Member Clusters access ASM data through a private network. The private network can be the same private network used by Grid Infrastructure or a separate dedicated *ASM private network*. This model of data access is useful for test and development configurations operating in Member Clusters for accessing production or test data that is shared through the DSC. The ASM Storage Network reduces the cost of storage with the use of network attachment rather than more expensive storage attachment, such as Fibre Channel switched networks.

The following slides focus on the second case, which is using ASM IO Server.

ASM IOServer Configuration



An ASM IOServer instance provides Oracle ASM file access for Oracle Database instances on nodes of Oracle member clusters that do not have connectivity to ASM managed disks.

In addition, IOServer enables you to configure client clusters on such nodes. On a storage cluster, an IOServer instance opens up the network ports to which clients send their IO. The IOServer instance receives data packets from the client and performs the appropriate IO to Oracle ASM disks, similar to any other database client. On the client side, databases can use dNFS to communicate with an IOServer instance. However, there is no client-side configuration, so you are not required to provide a server IP address or any additional configuration information. On nodes and clusters that are configured to access Oracle ASM files through IOServer, discovery of an Oracle IOServer instance occurs automatically.

The IOS Instance

An IOS instance contains the following processes:

- Network processes: These processes open up network ports and receive dNFS requests from IOS clients.
- Identifier processes: These processes pick up dNFS requests to create, delete, identify, un-identify, and resize ASM files.
- IO processes: These processes perform the actual IO on files that are identified by the local IOS instance.

ORACLE

The IOS instance contains the following processes:

- **Network processes:** These processes open up network ports and receive dNFS requests from the client side. These requests are then placed in a queue for IO and Identifier processes to pick up and process.
- **Identifier processes:** These processes pick up dNFS requests to create, delete, identify, un-identify, and resize Oracle ASM files.
- **IO processes:** These processes perform the actual IO on files that are identified by the local IOS instance.

The `ASM_IO_PROCESSES` initialization parameter specifies the number of IO worker processes to be started in an Oracle IO server instance.

The `ASM_IO_PROCESSES` Initialization Parameter

- The `ASM_IO_PROCESSES` parameter specifies the number of IO worker processes to be started in an IOS server instance.
- This parameter is applicable only for an ASM IOServer instance, which runs in an Oracle Grid Infrastructure home.
- You can control the number of IOServer instances running in the client cluster.

```
# srvctl modify ioserver -count 4

# srvctl config ioserver
ASM I/O Server instance count: 4
ASM I/O Server is enabled.
ASM I/O server is individually enabled on nodes:
ASM I/O server is individually disabled on nodes:
```


The `ASM_IO_PROCESSES` initialization parameter specifies the number of IO worker processes to be started in an ASM IOServer (IOS) server instance. This parameter is applicable only for an ASM IOServer instance, which runs in an Oracle Grid Infrastructure home.

IOServer instances are started automatically in the Oracle Grid Infrastructure configuration when generating the credential file for a client cluster. You can configure the number of IOServer instances running in the client cluster. The default value should work in most cases. However, under heavy IO loads, a higher value than the default may be appropriate.

Managing IOServer

- To add an Oracle ASM IOServer configuration to the cluster:

```
# srvctl add ioserver -count 3
```

- To enable or disable IOServer:

```
# srvctl enable ioserver
# srvctl disable ioserver -node host02
```

- To start or stop IOServer:

```
# srvctl start ioserver -node host02
# srvctl stop ioserver
```

- To relocate an IOServer instance from its current node:

```
# srvctl relocate ioserver -currentnode host01 -targetnode host02
```

- To display the status of IOServers running in a cluster:

```
# srvctl status ioserver -detail
```

ORACLE

When you configure, upgrade, or convert from a standard cluster to an Oracle Flex Cluster, you can create ASM IOServers on the Hub nodes in the cluster with the `srvctl add ioserver` command:

```
srvctl add ioserver [-spfile spfile] [-count ioserver_instance_count| ALL]
```

You can enable or disable IOServer for a cluster or on an individual node by using the `srvctl enable|disable ioserver` command:

```
srvctl enable|disable ioserver [-node node_name]
```

To start or stop IOServer clusterwide or on an individual node, use:

```
srvctl start|stop ioserver [-node node_name]
```

To relocate an Oracle ASM IOServer instance from its current hosting node to another node within the cluster, use the `srvctl relocate ioserver` command:

```
srvctl relocate ioserver -currentnode node_name [-targetnode node_name]
```

You can display the status of all Oracle ASM IOServers running in a cluster, the status of an Oracle ASM IOServer running on a particular Hub node, or information about clients from a specific database.

```
$ srvctl status ioserver -detail
ASM I/O Server is enabled
```

```
ASM I/O Server is running on nodes: host01, host02
ASM I/O Server instance +IOS1 running on node host01 is connected to ASM
instance
+ASM1:
Number of clients connected: 2
Names of clients: later14, alus16, later12, alus9, db0010
ASM I/O Server instance +IOS2 running on node host02 is connected to ASM
instance
+ASM2:
Number of clients connected: 3
Number of requests completed per sec: 110
Names of networks: eth1, eth2
Names of clients: later11, later3, later14, later15, later16, alus19,
alus17, alus13
```

Quiz

If you configure a four-node cluster with Flex ASM, which statement describes the resulting configuration?

a. ASM instances run on two cluster nodes for high availability, and more ASM instances are started as the number of ASM clients increases.

b. ASM instances run on the first three cluster nodes.

c. Three ASM instances are spread across the cluster.

d. Each of the four nodes runs an ASM instance.

Answer: c

The default cardinality for ASM instances is three. Regardless of the cluster size, Clusterware attempts to start three ASM instances when a new cluster is configured with Flex ASM. Fewer than three instances may start only if an error prevents an ASM instance from starting, or if there are fewer than three nodes in a standard cluster, or fewer than three Hub Nodes in a Flex Cluster. The ASM instances may start on the first three nodes, as suggested in answer *b*; however, this will not always be the case.

Quiz

Identify three statements that are true about server Flex ASM.

a. Flex ASM requires an ASM instance on each cluster node that is running an Oracle Database instance.

b. Flex ASM allows ASM clients to remotely connect to ASM over a network.

c. With Flex ASM, a small pool of ASM instances can be used to serve a larger pool of database servers.

d. If an ASM instance fails, the database clients and ASM cluster file systems can reconnect to another ASM instance.

Answer: b, c, d

Flex ASM relaxes the hard dependency between ASM and the database clients; therefore, it does not require an ASM instance on each cluster node that is running an Oracle Database instance.

Quiz

Which statement best describes the relationship between
Flex Clusters and Flex ASM?

a. There is no relationship, except that both have “Flex” in their names.
b. A Flex Cluster requires Flex ASM, but Flex ASM does not require a Flex Cluster.
c. Flex ASM requires a Flex Cluster, but a Flex Clusters does not require Flex ASM.
d. Flex Clusters and Flex ASM always require each other.

ORACLE

Answer: b

A Flex Cluster requires Flex ASM; however, Flex ASM can also run on a standard cluster providing I/O services on a subset of the cluster nodes.

Summary

In this lesson, you should have learned how to:

- Describe the architecture and components of Flex ASM
- Configure Flex ASM
- Administer Flex ASM

Practice 3 Overview: Database Fail Over with Flex ASM

In this practice, you crash an ASM instance and examine how the database client transparently fails over to another Flex ASM instance.

4

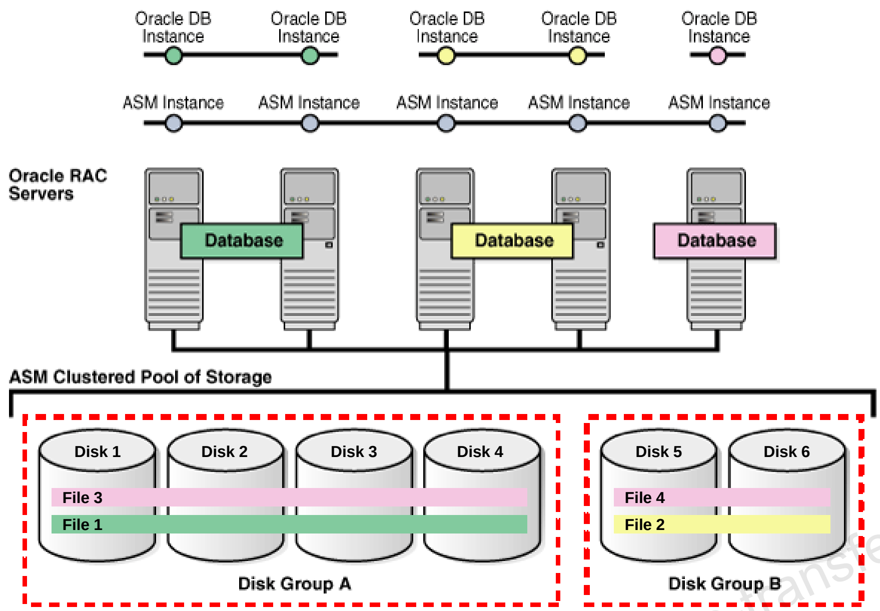
Administering ASM Disk Groups Part I

Objectives

After completing this lesson, you should be able to:

- Create and delete Automatic Storage Management (ASM) disk groups
- Set the attributes of an existing ASM disk group
- Perform ongoing maintenance tasks on ASM disk groups

Disk Group: Overview



A disk group is a grouping of one or more disks that ASM manages as a unit of storage. Each disk group contains the metadata that is required for management of space in the disk group. A disk group is conceptually similar to a logical volume (LV) in a typical Storage Area Network.

Files are allocated from the space inside a disk group. The content of the files that are stored in a disk group is evenly distributed, or striped, across the disks in the disk group to eliminate hot spots and to provide uniform performance across the disks.

ASM striping balances disk usage such that all the disks in a disk group are used evenly in percentage terms. When a disk group is made up of uniform-sized disks, the amount of data on each disk is approximately the same. When a disk group contains different-sized disks, the larger disks will contain more data than the smaller disks. A comparatively large disk may present an I/O bottleneck within a disk group if the bandwidth to all the disks in the disk group is the same. By default, ASM automatically rebalances storage whenever the storage configuration of a disk group changes such as when a disk is added.

Each ASM file is completely contained within a single disk group. However, a disk group can contain files belonging to several databases and a single database can use different files from multiple disk groups. For most installations, you need only a small number of disk groups.

Disk groups can be created by using ASMCA, ASMCMD, Enterprise Manager, SQL*Plus, and DBCA (when creating a database). In a clustered environment, a disk group resource is created when the disk group is created. Clusterware is responsible for starting and stopping the disk group resource. You can use `CRSCTL` or `SRVCLTL` to manage the disk group resource like any other Clusterware resource:

```
$ crsctl stop resource ora.DATA2.dg
CRS-2673: Attempting to stop 'ora.DATA2.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.DATA2.dg' on 'host02'
CRS-2677: Stop of 'ora.DATA2.dg' on 'host01' succeeded
CRS-2677: Stop of 'ora.DATA2.dg' on 'host02' succeeded

$ srvctl start diskgroup -diskgroup DATA2

$ crsctl stat res ora.DATA2.dg
NAME=ora.DATA2.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE          , ONLINE
STATE=ONLINE on host01, ONLINE on host02

$ srvctl status diskgroup -diskgroup DATA2
Disk Group DATA2 is running on host01,host02
```

Creating a New Disk Group with ASMCMD

ASMCMD can use XML-formatted input or an XML configuration file to create and change the disk group.

- Sample XML used with the `mkdg` command:

```
$ cat data.xml
<dg name="DATA" redundancy="normal">
   <fg name="fg1">
      <dsk string="/dev/sda1" />
      <dsk string="/dev/sdb1" />
   </fg>
   <fg name="fg2">
      <dsk string="/dev/sdc1" />
      <dsk string="/dev/sdd1" />
   </fg>
   <a name="compatible.asm" value="12.2"/>
   <a name="compatible.rdbms" value="12.2"/>
</dg>

$ asmcmd
ASMCMD> mkdg data.xml
```


ASMCMD has added the ability to use an XML configuration file to either create a disk group or change a disk group configuration.

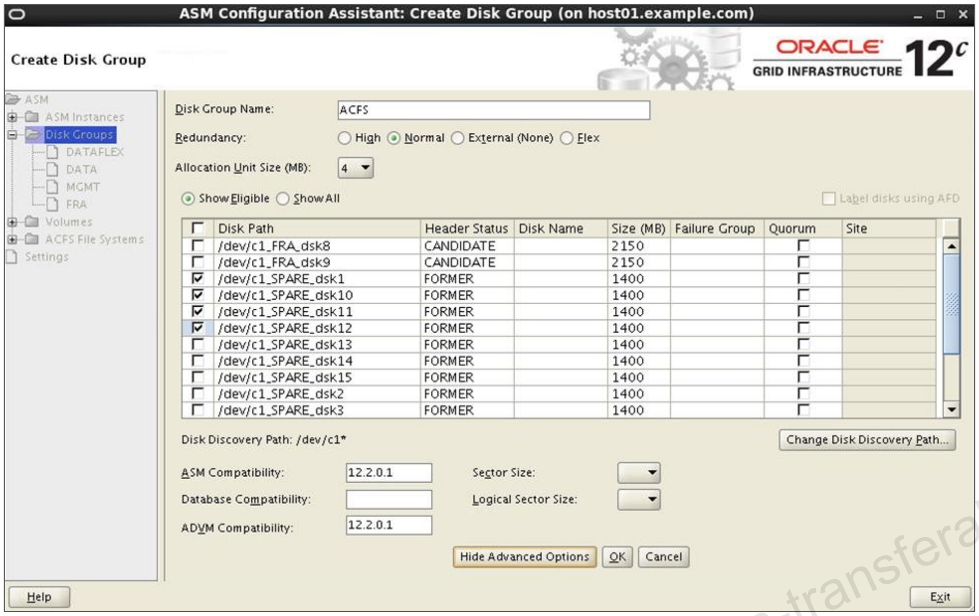
The XML file for the `mkdg` command specifies the name of the disk group, redundancy, attributes, and paths of the disks that form the disk group. Redundancy is an optional parameter; the default is normal redundancy. For some types of redundancy, the disks are required to be gathered into failure groups. In case the failure groups are not specified, every disk will be in its own failure group.

It is possible to set the disk group attribute values during disk group creation. Some attributes, such as `AU_SIZE` and `SECTOR_SIZE`, can be set only during disk group creation.

The following is an example of an inline XML configuration for `chdg`. This XML alters the disk group named `DATA`. The `FG1` failure group is dropped and the `DATA_0001` disk is also dropped. The `/dev/disk5` disk is added to the `FG2` failure group. The rebalance power level is set to `3`.

```
ASMCMD> chdg '<chdg> <dg name="DATA" power="3"> <drop> <fg name="FG1"> </fg>
<dsk name="DATA_0001" /> </drop> <add> <fg name="FG2"> <dsk
string="/dev/disk5"/> </fg> </add> </chdg>'
```

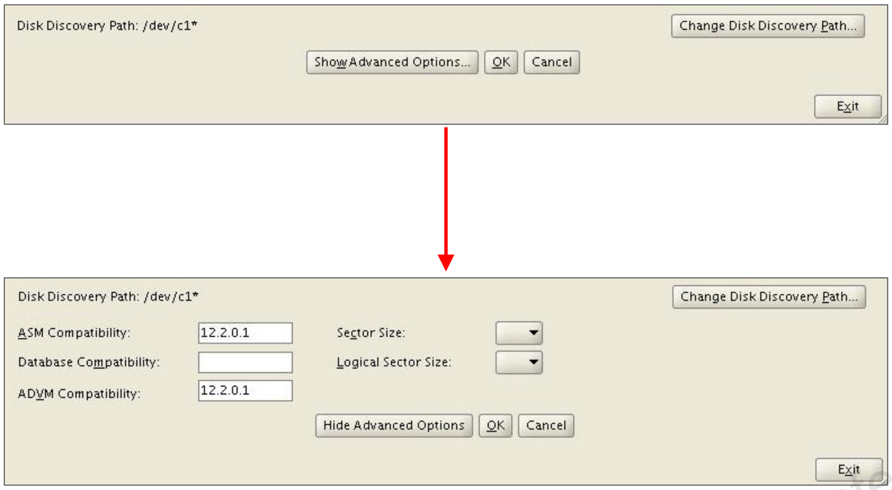
Creating an ASM Disk Group with ASMCA



In the slide, the ASMCA utility is being used to create a disk group. Redundancy is set to Normal. Four disks are selected to be members of this disk group: `c1_SPARE_dsk1`, `c1_SPARE_dsk10`, `c1_SPARE_dsk11`, and `c1_SPARE_dsk11`. The header status can show multiple valid values. If this is the first time that a disk has been created, the header status will be `CANDIDATE`. If the disk has been a member of the disk group and has been cleanly dropped, the header status will be `FORMER` as shown in the slide. The `PROVISIONED` header status is similar to `CANDIDATE` except that `PROVISIONED` implies that an additional platform-specific action has been taken by an administrator to make the disk available for ASM.

The disk discovery path is set by an initialization parameter, `ASM_DISKSTRING`. This parameter can consist of multiple search paths separated by commas. The default `ASM_DISKSTRING` parameter string is empty. There is an appropriate default discovery path for most OS platforms. A disk discovery string that limits the directories that are searched can reduce the time for discovery, but the disk discovery string must include all the disks that are members of existing disk groups.

Creating an ASM Disk Group: Advanced Options



At the bottom of the Create Disk Group page, click Show Advanced Options to see the portion of the page shown in the slide. In this portion of the page, you can set the disk group attributes: allocation unit size and compatibility parameters.

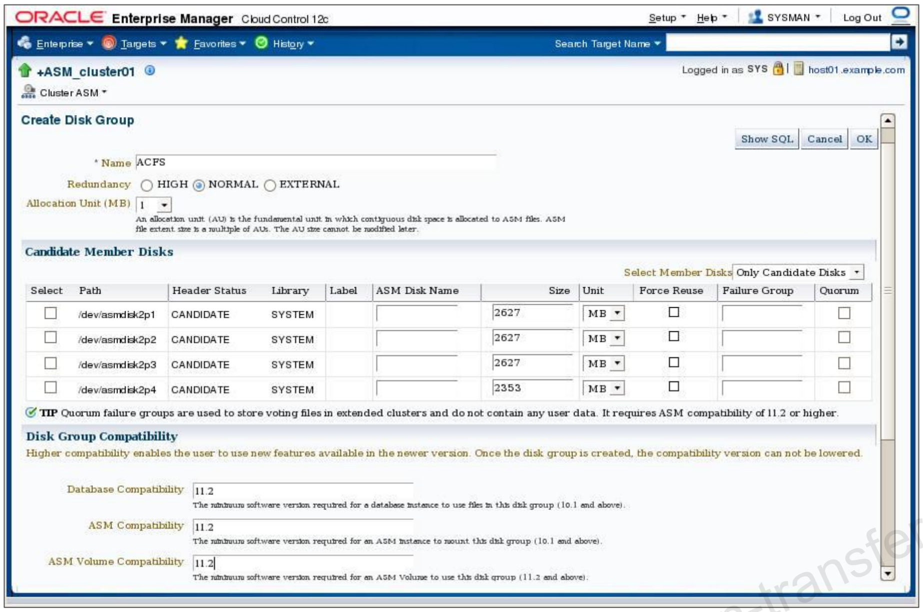
The field labeled ASM Compatibility sets the `COMPATIBLE.ASM` attribute. For Oracle ASM in Oracle Database 12*c*, 12.2 is the default setting for the `COMPATIBLE_ASM` attribute when you use the SQL `CREATE DISKGROUP` statement, the ASMCMD `mkdg` command, and ASMCA. When setting values for compatibility attributes, specify at least the major and minor versions of a valid Oracle Database release number. For example, you can specify compatibility as 12.1 or 12.2; Oracle assumes that any missing version number digits are zeroes.

Database Compatibility sets the minimum version level for any database instance that is allowed to use the disk group. This is the `COMPATIBLE.RDBMS` attribute. For Oracle ASM in Oracle Database 12*c*, 10.1 is the default setting when you use the SQL `CREATE DISKGROUP` statement, the ASMCMD `mkdg` command, and ASMCA.

To use ADVM volumes, ADVM Compatibility must be set to 11.2 or later and ASM Compatibility must be 11.2.or later. ADVM Compatibility sets the `COMPATIBLE.ADVM` attribute. By default, the value of the `COMPATIBLE.ADVM` attribute is empty until it is set.

Note: Advancing the values for disk group compatibility attributes is an irreversible operation.

Creating a Disk Group with Enterprise Manager



ORACLE

The same functionality is available with Enterprise Manager. Just as with the ASMCA utility, you can specify the disks and failure groups on the Create Disk Group page in Enterprise Manager.

Creating a Disk Group with SQL*Plus

The `CREATE DISKGROUP` command creates ASM disk groups.

```
CREATE DISKGROUP diskgroup_name
  [ { HIGH | NORMAL | EXTERNAL } REDUNDANCY ]
  { [ FAILGROUP failgroup_name ]
    DISK qualified_disk_clause [, qualified_disk_clause]...
  } ...
  [ ATTRIBUTE { 'attribute_name' = 'attribute_value' }... ]
;

qualified_disk_clause ::= search_string
  [ NAME disk_name ]
  [ SIZE size_clause ]
  [ FORCE | NOFORCE ]
```

Example:

```
CREATE DISKGROUP FRA NORMAL REDUNDANCY
DISK 'ORCL:SDD11' NAME 'FRA_DISK1' SIZE 977 M,
     'ORCL:SDD12' NAME 'FRA_DISK2' SIZE 977 M;
```

ORACLE

The `CREATE DISKGROUP` statement creates a disk group, assigns one or more disks to the disk group, and mounts the disk group for the first time. If you want ASM to automatically mount the disk group when an ASM instance starts, you must add the disk group name to the value of the `ASM_DISKGROUPS` initialization parameter in your parameter files (PFILEs). If you use a server parameter file (SPFILE), the disk group is added to the initialization parameter automatically.

The `CREATE DISKGROUP` command can also be run by using database management tools such as ASM Configuration Assistant (ASMCA), Oracle Enterprise Manager, and the ASM Command-Line utility (ASMCMD).

The `CREATE DISKGROUP` statement has the following clauses:

`REDUNDANCY` Clause

The `REDUNDANCY` clause allows you to specify the redundancy level of the disk group.

`NORMAL REDUNDANCY` requires the existence of at least two failure groups. By default, `NORMAL REDUNDANCY` provides a two-way mirror of all ASM files except control files, which are mirrored three ways. `NORMAL REDUNDANCY` disk groups can tolerate the loss of one failure group.

HIGH REDUNDANCY requires the existence of at least three failure groups. ASM fixes mirroring at three-way mirroring, with each file getting two mirrored copies. HIGH REDUNDANCY disk groups can tolerate the loss of two failure groups.

EXTERNAL REDUNDANCY indicates that ASM does not provide any redundancy for the disk group. The disks within the disk group must provide redundancy (for example, by using a storage array), or you must be willing to tolerate the loss of the disk group if a disk fails. You cannot specify the FAILGROUP clause if you specify EXTERNAL REDUNDANCY.

FAILGROUP Clause

Use this clause to specify a name for one or more failure groups. If you omit this clause, and you have specified NORMAL or HIGH REDUNDANCY, ASM automatically adds each disk in the disk group to its own failure group. The implicit name of the failure group is the same as the name in the NAME clause.

DISK Clause

Use this clause to specify one or more disks for each failure group.

For each disk that you are adding to the disk group, specify the operating system–dependent search string that ASM will use to find the disk. The search_string must point to a subset of the disks returned by discovery by using the strings in the ASM_DISKSTRING initialization parameter. If search_string does not point to any disks to which the ASM user has read/write access, ASM returns an error. If it points to one or more disks that have already been assigned to a different disk group, Oracle Database returns an error unless you also specify FORCE. For each valid candidate disk, ASM formats the disk header to indicate that it is a member of the new disk group.

The optional NAME subclause is valid only if search_string points to a single disk. It specifies an operating system–independent name for the disk. The name can be up to 30 alphanumeric characters. The first character must be alphabetic. If you omit this clause and you assigned a label to a disk through ASMLib, that label is used as the disk name. If you are not using ASMLib, ASM creates a default name of the form diskgroup_name_*nnnn*, where *nnnn* is the disk number. You can use this name to refer to the disk in subsequent ASM operations.

Use the optional SIZE subclause to specify the size of the disk. If you specify a size greater than the capacity of the disk, ASM returns an error. If you specify a size less than the capacity of the disk, you limit the disk space that ASM will use. If you omit this clause, ASM attempts to determine the size of the disk programmatically.

You can specify FORCE or NOFORCE for each disk.

Specify FORCE if you want ASM to add the disk to the disk group even if the disk is already a member of a different disk group. Exercise caution because using FORCE thus may destroy existing disk groups. For this clause to be valid, the disk must already be a member of a disk group and the disk cannot be part of a mounted disk group.

Specify NOFORCE if you want ASM to return an error if the disk is already a member of a different disk group. NOFORCE is the default.

ATTRIBUTE Clause

Use this clause to set attribute values for the disk group. ASM disk group attributes are described later in this lesson.

Disk Group Attributes

The following attributes can be set for a disk group:

Attribute	Description	Valid Values	Default Value
AU_SIZE	Specifies the AU size. The attribute can be set only during disk group creation.	1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, 64 MB	4 MB
DISK_REPAIR_TIME	Specifies the amount of time ASM will wait from when a disk goes offline until ASM drops it and rebalances the disk group	0 to 136 years specified in minutes (M) or hours (H)	3.6H
COMPATIBLE.RDBMS	Specifies the minimum software version that is required for a database instance to use files in this disk group	At least the first two digits of a valid Oracle Database release number	10.1
COMPATIBLE.ASM	Specifies the minimum software version that is required for an ASM instance to mount this disk group	At least the first two digits of a valid Oracle Database release number	12.2
COMPATIBLE.ADVM	Allows creation of ASM volumes	>=11.2	NONE

ORACLE

Disk group attributes govern various aspects of how a disk group functions.

AU_SIZE specifies the allocation unit size, which is the fundamental unit of space within a disk group. It is also the stripe size for coarse-grained striping. This attribute can be set only during disk group creation.

DISK_REPAIR_TIME specifies the amount of time ASM will wait from when a disk goes offline until ASM drops it and rebalances the disk group. This attribute enables the fast mirror resync feature, whereby ASM keeps track of pending changes on an offline disk during an outage and the extents are automatically resynced when the disk is brought back online.

There are three compatibility attributes. COMPATIBLE.RDBMS specifies the minimum software version that is required for a database instance to use files in this disk group, whereas COMPATIBLE.ASM specifies the minimum software version that is required for an ASM instance to mount this disk group. The compatibility attributes are discussed in greater detail later in this lesson. The value for the disk group **COMPATIBLE.ADVM** attribute determines whether the disk group can contain Oracle ASM volumes. COMPATIBLE.ASM must be 11.2 or later first, and the ASM Dynamic Volume Manager (ADVM) drivers must be loaded.

Disk Group Attributes

The following attributes can be set for a disk group:

Attribute	Description	Valid Values	Default Value
CONTENT.CHECK	Enables or disables content checking when performing data copy operations for rebalancing	TRUE or FALSE	FALSE
CONTENT.TYPE	Identifies the disk group type	DATA, SYSTEM, or RECOVERY	DATA
SECTOR_SIZE	Is used with CREATE DISKGROUP to specify the disk sector size	512, 4096, 4K	Platform dependent
LOGICAL_SECTOR_SIZE	specifies the smallest unit of I/O allowed on the disk group.	512, 4096, 4K	Platform dependent
PREFERRED_READ.ENABLED	Controls whether preferred read functionality is enabled for a disk group.	TRUE or FALSE	FALSE
STORAGE_TYPE	Specifies the type of the disks in the disk group	AXIOM, ZFSSA, and OTHER	None

ORACLE

CONTENT.CHECK enables or disables content checking when performing data copy operations for rebalancing a disk group. The attribute value can be set to true or false. The content checking can include Hardware Assisted Resilient Data (HARD) checks on user data, validation of file types from the file directory against the block contents and file directory information, and mirror side comparison. When the attribute is set to true, logical content checking is enabled for all rebalance operations.

CONTENT.TYPE identifies the disk group type: data, recovery, or system. The type value determines the distance to the nearest neighbor disk in the failure group where ASM mirrors copies of the data. ASM uses this attribute value to make it less likely that a double failure in the storage medium causes disk groups of different content types to become unavailable. The default value is `data`, and specifies a distance of 1 to the nearest neighbor disk. A value of `recovery` specifies a distance of 3 and a value of `system` specifies a distance of 5. The attribute is valid only for disk groups that are set to normal or high redundancy. `COMPATIBLE.ASM` must be set to 11.2.0.3 or later.

SECTOR_SIZE specifies the sector size for the disks in a disk group and can be set only when creating a disk group. The values can be set to 512, 4096, or 4K if the disks support these values. The default value is platform-dependent.

The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` disk group attributes must be set to 11.2 or later to set the sector size to a value other than the default value.

LOGICAL_SECTOR_SIZE specifies the logical sector size for a disk group. The value can be set when creating or altering a disk group. The value of LOGICAL_SECTOR_SIZE specifies the smallest unit of I/O allowed on the disk group. The value must be less than or equal to the size of the SECTOR_SIZE disk group attribute. COMPATIBLE.ASM must be set to 12.2 or greater to use the LOGICAL_SECTOR_SIZE disk group attribute.

PREFERRED_READ.ENABLE controls whether preferred read functionality is enabled for a disk group.

If preferred read functionality is enabled, then this functionality enables an instance to determine and read from disks at the same site as itself, which can improve performance. Each Oracle ASM instance in an Oracle extended cluster has the information to determine the site in which it is located. If preferred read functionality is enabled, then based on the site information, the preferred status for a disk is determined to be true if the disk is in the same site as the instance. Otherwise, if the disk is not located at the same site as the Oracle ASM instance, the preferred status for a disk is set to false.

For extended clusters, the preferred read setting is enabled by default with a TRUE setting for the PREFERRED_READ.ENABLED disk group attribute of each disk group. For clusters that are not extended (only one physical site), preferred read is disabled. Preferred read status applies to extended, normal, high, and flex redundancy disk groups.

Oracle Database 12*c* Release 2 (12.2) is required and COMPATIBLE.ASM must be set to 12.2 or greater to use PREFERRED_READ.ENABLED.

Whether or not PREFERRED_READ.ENABLED has been enabled, preferred read can be set at the failure group level on an Oracle ASM instance or a client instance in a cluster with the ASM_PREFERRED_READ_FAILURE_GROUPS initialization parameter, which is available for backward compatibility.

To determine whether a disk has preferred read status, you can check the PREFERRED_READ column of the V$ASM_DISK view.

STORAGE_TYPE specifies the type of the disks in the disk group. The possible values are AXIOM, ZFSSA, and OTHER. If the attribute is set to AXIOM or ZFSSA, all disks in the disk group must be of that type. If the attribute is set to OTHER, any types of disks can be in the disk group. If the STORAGE.TYPE disk group attribute is set to AXIOM or ZFSSA, functionality for Hybrid Columnar Compression (HCC) can be enabled for Pillar Axiom or ZFS storage. To set the STORAGE.TYPE attribute, the COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes must be set to 11.2.0.3 or later.

Disk Group Attributes

The following attributes can be set for a disk group:

Attribute	Description	Valid Values	Default Value
ACCESS_CONTROL.ENABLED	Allows the creation of access control lists (ACL)	TRUE, FALSE	FALSE
ACCESS_CONTROL.UMASK	Sets the default permissions to be set for files created in the disk group	0, 2, 6 for owner, group, and others	066
CELL.SMART_SCAN_CAPABLE	Enables smart scan predicate offload processing if all disk group disks are Exadata Grid Disks	TRUE, FALSE	FALSE
FAILGROUP_REPAIR_TIME	Specifies a default repair time for the failure groups in the disk group		24 Hours
THIN_PROVISIONED	Enables or disables the functionality to discard unused storage space after a rebalance	TRUE, FALSE	FALSE

To manage Oracle ASM File Access Control for a disk group, you must set the **ACCESS_CONTROL.ENABLED** and **ACCESS_CONTROL.UMASK** disk group attributes. Before setting the ACCESS_CONTROL.UMASK disk group attribute, you must set the ACCESS_CONTROL.ENABLED attribute to true to enable Oracle ASM File Access Control.

The **CELL.SMART_SCAN_CAPABLE** attribute enables smart scan predicate offload processing if all the disks in the disk group are Exadata Grid Disks.

FAILGROUP_REPAIR_TIME specifies a default repair time for the failure groups in the disk group. Failure group repair time is used if Oracle ASM determines that an entire failure group has failed. The default value is 24 hours (24h). If repair time is specified for a disk, such as with the DROP AFTER clause of the ALTER DISKGROUP OFFLINE DISK statement, that disk repair time overrides the failure group repair time.

This attribute can be set only when altering a disk group and is applicable only to normal and high redundancy disk groups.

THIN_PROVISIONED enables or disables the functionality to discard unused storage space after a disk group rebalance is completed. The attribute value can be true or false. The default value is false. Storage vendor products that support thin provisioning have the capability to reuse the discarded storage space for more efficient overall physical storage utilization.

For more information about ASM Disk Group attributes, refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 2 (12.2)*

Viewing Disk Group Attributes

- In an ASM instance, V$ASM_ATTRIBUTE displays one row for each attribute that is defined.
- In addition to the attributes specified by the CREATE DISKGROUP and ALTER DISKGROUP statements, the view may show other attributes that are created automatically.
- Disk group attributes are listed in V$ASM_ATTRIBUTE only if the disk group attribute COMPATIBLE.ASM is set to 11.1 or later.
- The same information can be displayed by using the ASMCMD LSATTR command:

```
$ asmcmd lsattr -lm -G <diskGroup>
```


You can display disk group attributes with the V$ASM_ATTRIBUTE view and the ASMCMD lsattr command. V$ASM_ATTRIBUTE lists the ASM disk group attributes if the disk group attribute COMPATIBLE.ASM is set to 11.1 or later. If COMPATIBLE.ASM is not set to 11.1 or later, V$ASM_ATTRIBUTE will have no rows corresponding to the disk group. An example of a query that retrieves attributes for a particular disk group is as follows:

```
SELECT G.NAME DISK_GROUP, A.NAME ATTRIBUTE, A.VALUE FROM V$ASM_ATTRIBUTE A,
V$ASM_DISKGROUP G WHERE A.GROUP_NUMBER = G.GROUP_NUMBER AND G.NAME = 'DATA';
```

The ASMCMD lsattr -lm -G command will show the same information for a disk group.

```
ASMCMD> lsattr -lm -G DATA

Group_Name  Name                     Value       RO  Sys
DATA        access_control.enabled   FALSE       N   Y
DATA        access_control.umask     066         N   Y
DATA        au_size                  1048576     Y   Y
DATA        cell.smart_scan_capable  FALSE       N   N
DATA        compatible.advm          12.2.0.0.0  N   Y
DATA        compatible.asm           12.2.0.0.0  N   Y
DATA        compatible.rdbms         12.2.0.0.0  N   Y
DATA        content.check            FALSE       N   Y
DATA        content.type             data        N   Y
DATA        disk_repair_time         3.6h        N   Y
DATA        failgroup_repair_time    24.0h       N   Y
...
```

Compatibility Attributes

- Disk group compatibility attributes can be set by using `CREATE DISKGROUP` or `ALTER DISKGROUP`.
- Values can only be advanced and the setting is irreversible.
- `COMPATIBLE.RDBMS` must be less than or equal to `COMPATIBLE.ASM`.
- Some valid attribute combinations:

COMPATIBLE.ASM	COMPATIBLE.RDBMS	COMPATIBLE.ADVM	Oracle ASM Instance Version	COMPATIBLE Setting for RDBMS Instance
10.1	10.1	n/a	>= 10.1	>= 10.1
11.1	10.1	n/a	>= 11.1	>= 10.1
11.2	11.2	11.2	>= 11.2	>= 11.2
12.1	12.1	12.1	>= 12.1	>= 12.1
12.2	12.1	12.2	>= 12.2	>= 12.1

ORACLE

The matrix table in the slide shows some possible combinations of Oracle ASM and database releases:

- The database release is Oracle Database 11*g* Release 2 (11.2) (database initialization parameter `COMPATIBLE` is set to 11.2) and the Oracle ASM release is Oracle ASM 11*g* Release 2 (11.2). The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` disk group attributes are set to 10.2 for a disk group. The Oracle ASM disk group functionality remains at Oracle Database 10*g* Release 2 (10.2).
- The database release is Oracle Database 10*g* Release 1 (10.1) and the Oracle ASM release is 11.2. `COMPATIBLE.ASM` is set to 11.2 and `COMPATIBLE.RDBMS` is set to 10.1 for a disk group.
- The database release is Oracle Database 12*c* Release 2 (12.2) (database initialization parameter `COMPATIBLE` is set to 12.2) and the Oracle ASM release is Oracle ASM 12*c* Release 2 (12.2). All the disk group compatibility attributes are set to 12.2 for a disk group.

The Oracle ASM features enabled for this combination of attribute settings are shown in the next slide.

Features Enabled by Disk Group Compatibility Attributes

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
ASM_POWER_LIMIT value up to 1024	>= 11.2.0.2	n/a	n/a
Content type of a disk group	>= 11.2.0.3	n/a	n/a
Replication status of a disk group	>= 12.1	n/a	n/a
Managing a shared password file in a disk group	>= 12.1	n/a	n/a

ORACLE

The table in the slide describes the Oracle ASM features enabled by valid combinations of the disk group compatibility attribute settings.

The following list applies to the table.

- The value of COMPATIBLE.ASM must always be greater than or equal to the value of COMPATIBLE.RDBMS and COMPATIBLE.ADVM.
- A value of not applicable (n/a) means that the setting of the attribute has no effect on the feature.
- Oracle ASM features not explicitly listed do not require advancing the disk group compatibility attribute settings.
- Oracle ASM features explicitly identified by an operating system are available on that operating system starting with the associated disk group attribute settings.
- If encryption is configured for the first time on Oracle ASM 11*g* Release 2 (11.2.0.3) on Linux or if encryption parameters must be changed or a new volume encryption key must be created following a software upgrade to Oracle ASM 11*g* Release 2 (11.2.0.3) on Linux, then the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or higher.

For information about Oracle ACFS features enabled by disk group compatibility attribute settings, refer to *Oracle Automatic Storage Management Administrator’s Guide, 12c Release 2 (12.2)*

Features Enabled by Disk Group Compatibility Attributes

Disk Group Features Enabled	COMPATIBLE.ASM	COMPATIBLE.RDBMS	COMPATIBLE.ADVM
Greater than 2 TB Oracle ASM disks without Oracle Exadata storage	>= 12.1	>= 12.1	n/a
Appliance mode support	>= 12.1.0.2	n/a	n/a
LOGICAL_SECTOR_SIZE	>= 12.2	n/a	n/a
Altering sector size	>= 12.2	n/a	n/a
Oracle ASM flex and extended disk groups	>= 12.2	>= 12.2	n/a
SCRUB_ASYNC_LIMIT	>= 12.2	n/a	n/a
PREFERRED_READ.ENABLED Oracle Database 12c Release 2 (12.2) is required.	>= 12.2	n/a	n/a

The table in the slide describes the Oracle ASM features enabled by valid combinations of the disk group compatibility attribute settings.

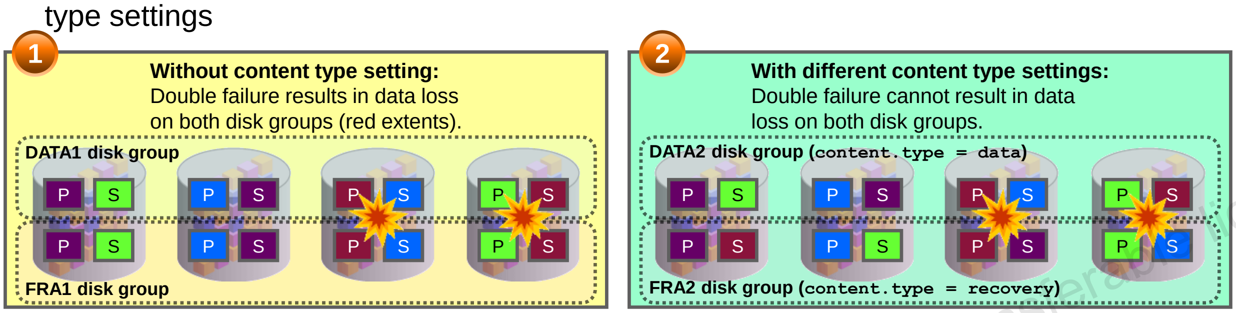
Specifying Content Type for a Disk Group

Administrators can specify the content type for each disk group:

- New disk group attribute: **`content.type`**
  - Possible values: `data`, `recovery`, or `system`
  - Configuration example:

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'content.type'='data';
```

- Decreased likelihood that multiple failures would impact disk groups with different content type settings



For `NORMAL` and `HIGH` redundancy ASM disk groups, the algorithm that determines the placement of secondary extents (mirror copies of data) uses an adjacency measure to determine the placement.

In earlier versions of ASM, the same algorithm and adjacency measure were used for all disk groups.

With Oracle Database 12*c*, ASM provides administrators with the option to specify the content type associated with each ASM disk group. This capability is provided by a new disk group attribute, `CONTENT.TYPE`. Three possible settings are allowed: `data`, `recovery`, or `system`. Each content type setting modifies the adjacency measure used by the secondary extent placement algorithm.

The result is that the contents of disk groups with different content type settings are distributed differently across the available disks. This decreases the likelihood that a double failure will result in data loss across multiple `NORMAL` redundancy disk groups with different content type settings. Likewise, a triple failure is less likely to result in data loss on multiple `HIGH` redundancy disk groups with different content type settings.

To illustrate, consider the diagram at the bottom of the slide. Example 1 shows two `NORMAL` redundancy disk groups, `DATA1` and `FRA1`, which are configured without the content type setting. Both disk groups use the same algorithm for placing secondary extents.

That is, the secondary extent is placed on the disk immediately to the right of the disk containing the primary extent. Where the primary extent is on the far-right disk, the secondary extent is placed on the far-left disk.

In this example, failure of the two disks at the extreme right results in data loss in both disk groups; that is, the red extents. In this case, it is possible that a double failure could result in the loss of a data file and the archived log files required to recover it.

Example 2 shows `NORMAL` redundancy disk groups, `DATA2` and `FRA2`, which are configured with different content type settings. In this example, the `DATA2` disk group uses the same placement algorithm as before. However, the data placement for `FRA2` uses a different adjacency measure, and because of this, the contents of `FRA2` are spread differently across the disks.

In this example, failure of the two disks at the extreme right results in data loss only in the `DATA2` disk group. However, because of different distribution of data that is associated with the different content type setting, `FRA2` experiences no data loss. In this case, a double failure might result in the loss of a data file, but the archived log files required to recover it are still available.

Note that the diagrams and associated examples described here are illustrative only. The actual placement algorithm is more involved, and each disk is typically partnered with more than one other disk.

Note also that the content type attribute setting does not govern the actual contents of the disk group. That is, any type of file can be located on any disk group regardless of the content type setting. For example, a disk group with `content.type=data` can store the flash recovery area for an Oracle database. Likewise, another disk group with `content.type=recovery` can be used to store database data files. It remains the responsibility of the ASM administrator to ensure that each file is located in the appropriate disk group.

Support for 4 KB Sector Disk Drives

- Oracle ASM provides support for 4 KB sector disk drives without negatively affecting performance.
- The values for `SECTOR_SIZE` can be set to 512, 4096, or 4K if the disks support these values.
  - The default value is platform-dependent.
- In native mode, there is no performance penalty with:
  - ASM files, at the disk group level
- The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` attributes must be set to 11.2 or later to set the sector size to a value other than the default value.
- There is a performance penalty for ACFS when using 4 KB sector disk drives in 512 sector emulation mode.

ORACLE

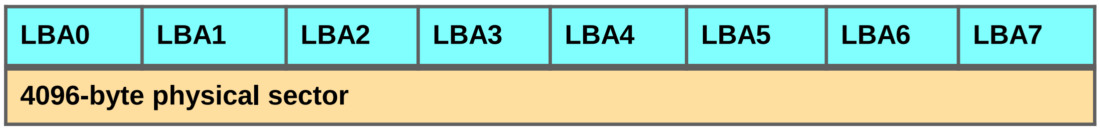
You can use the optional `SECTOR_SIZE` disk group attribute with the `CREATE DISKGROUP` SQL statement to specify disks with the sector size set to the value of `SECTOR_SIZE` for the disk group. Oracle ASM provides support for 4 KB sector disk drives without negatively affecting performance. The `SECTOR_SIZE` disk group attribute can be set only during disk group creation.

`SECTOR_SIZE` can be set to 512, 4096, or 4K if the disks support these values. The default value is platform-dependent. The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` disk group attributes must be set to 11.2 or higher to set the sector size to a value other than the default value.

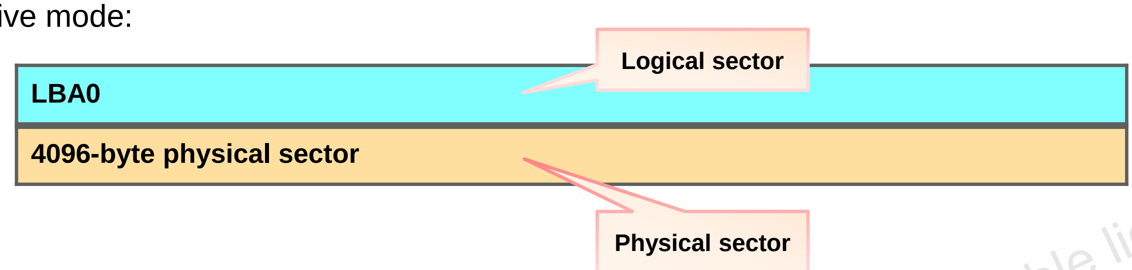
Oracle ACFS does not support 4 KB sector drives directly. There is a performance penalty for Oracle ACFS when using 4 KB sector disk drives in 512 sector emulation mode.

Supporting 4 KB Sector Disks

- Emulation mode:



- Native mode:



4 KB sector disks have physical sectors (shown in light yellow) and logical sectors (shown in aqua blue). There are two types of 4 KB sector disks: emulation mode and native mode.

- 4 KB sector disks in emulation mode have eight logical sectors per physical sector (as shown in the slide). They maintain a 512-byte interface to their 4 KB physical sectors—that is, the logical block address (LBA) references 512 bytes on disk.
- 4 KB sector disks in native mode have one logical sector per physical sector (as shown in the slide). So, there is only the 4 KB interface. In other words, the LBA references 4096 bytes on disk.

Emulation mode can affect performance because the disk drive reads the 4 KB sector into disk cache memory, changes the 512-byte section, and then writes the entire 4 KB sector back to disk. For example, when redo is being written to disk in 512 chunks, each write requires a read of the 4 KB sector, an update of the 512-byte section, and then a write. With native mode, the 4 KB sector is written without requiring a read and an update.

ASM Support for 4 KB Sector Disks

- ASM commands:
  - SQL: `CREATE DISKGROUP…ATTRIBUTE SECTOR_SIZE`
  - ASMCMD MKDG :`<a name="sector_size" value="val"/>`
- ASMCA support
- Enterprise Manager support
- Requirement:
  - All the disks in a disk group must have the same sector size.
- On redo log files:
  - The `BLOCKSIZE` redo log file and the `SECTOR_SIZE` disk group must match.

ASM sector size can be managed by using three different tools: SQL command line, the ASM Command-Line utility (ASMCMD), and Enterprise Manager. Because disk groups must have a compatible sector size on all the disks in a disk group, sector size is set as an attribute of the disk group when the disk group is created. The `ALTER DISKGROUP ADD DISK` command requires that the candidate disk to be added has the same sector size as the existing disks in the group. (This should be handled by checking disk properties at the time the `ADD DISK` command is executed.)

Best Practice: For data files and temp files, avoid using data block size of less than 4 KB with 4 KB sector disk drives. The default data block size is 8 KB.

For redo log files and archive log files, the block size specified for the log file must match the sector size of the disk group. The syntax for specifying a new log file in the `DATA` disk group of 100 MB with the redo block size of 4 KB is:

```
SQL> ALTER DATABASE ADD LOGFILE +DATA SIZE 100M BLOCKSIZE 4K;
```

Sector Size Validations

- ASM prevents disks of different sector sizes from being added to the same disk group.
  - Validation occurs during `CREATE DISKGROUP`, `ALTER DISKGROUP ADD DISK`, and `ALTER DISKGROUP MOUNT` operations.
- If `SECTOR_SIZE` is explicitly specified, ASM attempts to verify that all the disks discovered through disk search strings have a sector size equal to the specified value.
- ASM attempts to verify disk sector size during the mount operation and fails if one or more disks have a sector size different than the value of the `SECTOR_SIZE` attribute.
- If `SECTOR_SIZE` is not specified and ASM can verify that all discovered disks have the same sector value, that value is assumed for the disk group sector size that is created.

ORACLE

The following validations apply to the sector size disk group attribute.

- ASM prevents disks of different sector sizes from being added to the same disk group. This validation occurs during `CREATE DISKGROUP`, `ALTER DISKGROUP ADD DISK`, and `ALTER DISKGROUP MOUNT` operations.
- If the `SECTOR_SIZE` attribute is explicitly specified when creating a disk group, ASM attempts to verify that all the disks that are discovered through disk search strings have a sector size that is equal to the specified value. If one or more disks have a sector size different from the specified value, or if ASM was not able to verify a disk sector size, the create operation fails.
- ASM also attempts to verify disk sector size during the mount operation and the mount fails if one or more disks have a sector size that is different from the `SECTOR_SIZE` value.
- If the `SECTOR_SIZE` attribute is not specified when creating a disk group and ASM can verify that all discovered disks have the same sector value, that value is assumed for the disk group sector size that is created. If the disks have different sector sizes, the create operation fails.
- When new disks are added to an existing disk group by using the `ALTER DISKGROUP .. ADD DISK` SQL statement, you must ensure that the new disks to be added have the same value as the `SECTOR_SIZE` disk group attribute. If the new disks have different sector sizes, the alter operation fails.

Using the SECTOR_SIZE Clause

Creating a disk group (in ASM) with a 4 KB sector size:

```
CREATE DISKGROUP mydgroup1 NORMAL REDUNDANCY
FAILGROUP mycontroller1 DISK
'/devices/diska1',
'/devices/diska2',
'/devices/diska3',
'/devices/diska4'
FAILGROUP mycontroller2 DISK
'/devices/diskb1',
'/devices/diskb2',
'/devices/diskb3',
'/devices/diskb4'
ATTRIBUTE 'compatible.asm'='12.2','compatible.rdbms'='12.2', 'sector_size'='4096';
```

ORACLE

Use the optional SECTOR_SIZE clause in the CREATE DISKGROUP command to explicitly specify a sector size value for the disk group. If you do not use this clause, ASM detects the hardware sector size and uses it. The SECTOR_SIZE disk group attribute can be set only during disk group creation. SECTOR_SIZE can be set to 512, 4096, or 4K. ASM provides support for 4 KB sector disk drives without negatively affecting performance. The following example shows the contents of an XML file that is used to create a disk group by specifying SECTOR_SIZE with the ASMCMD MKDG command.

```
<dg name="DATA2" redundancy="normal">
  <fg name="fg1">
    <dsk string="/dev/asmdisk2p1"/>
   </fg>
  ...
  <a name="sector_size" value="4096"/>
</dg>
```

You can determine the sector size value that has either been assumed or explicitly set for successful disk group creation by querying the V$ASM_ATTRIBUTE view. You can also query the SECTOR_SIZE column in the V$ASM_DISKGROUP view. The asmcmd lsattr command also displays the sector size of a disk group.

ASM Extended Support for 4K Sector Size

- 4K sector size support is provided by the new disk group attribute `LOGICAL_SECTOR_SIZE`.
  - This attribute specifies the logical sector size of the disk group.
  - The value specifies the smallest possible I/O the disk group can accept and limits the kind of disks that can join the disk group.
- The `LOGICAL_SECTOR_SIZE` column is now included in the `V$ASM_DISKGROUP` and `V$ASM_DISK` views.

```
SQL> select name, sector_size, logical_sector_size from V$ASM_DISKGROUP;

NAME                           SECTOR_SIZE  LOGICAL_SECTOR_SIZE
------------------------------ ----------- -------------------
DATA                                   512                4096
MGMT                                   512                4096
```

ORACLE

The new disk group attribute, `LOGICAL_SECTOR_SIZE`, sets the logical sector size of the disk group instead of the size that is estimated from the disks that are added to the disk group. The value specifies the smallest possible I/O that the disk group can accept, and also limits the kind of disks that can join the disk group. A new column, `LOGICAL_SECTOR_SIZE`, is included in the `V$ASM_DISKGROUP` and `V$ASM_DISK` views. This column represents the logical sector size value of the disk group or disk in bytes.

Specifying Disk Logical and Sector Sizes in a Disk Group

- Use LOGICAL_SECTOR_SIZE and SECTOR_SIZE with CREATE DISKGROUP or ALTER DISKGROUP:

```
CREATE DISKGROUP data1 NORMAL REDUNDANCY
  FAILGROUP controller1 DISK
    '/devices/asmdiskC1',
    ...
'sector_size'='4096', 'logical_sector_size'='512';
```

```
ALTER DISKGROUP data2
SET ATTRIBUTE 'sector_size'='4096', 'logical_sector_size'='4096';
```

- The disks specified must support sector size emulation.
- The LOGICAL_SECTOR_SIZE disk group attribute can be set to 512, 4096, or 4K.
- The COMPATIBLE.ASM disk group attribute must be set to 12.2 or later to set the logical sector size.

You can use the optional LOGICAL_SECTOR_SIZE and SECTOR_SIZE disk group attributes with the CREATE DISKGROUP and ALTER DISKGROUP SQL statements to specify a logical sector size and physical sector size for a disk group. These disks must support sector size emulation.

The SECTOR_SIZE disk group attribute can be set to 512, 4096, or 4 K if the disks support these values. The default value is platform-dependent. The COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes must be set to 11.2 or later to enable setting a value for sector size during disk group creation. The COMPATIBLE.ASM disk group attribute must be set to 12.2 or later to alter the sector size.

The LOGICAL_SECTOR_SIZE disk group attribute can explicitly set the logical sector size of a disk group instead of the size that is estimated from the disks that join the disk group. This attribute can be set to 512, 4096, or 4 K. The value specifies the smallest possible I/O that the disk group can accept, and also limits the types of disks that can join the disk group. The COMPATIBLE.ASM disk group attribute must be set to 12.2 or later to set the logical sector size during the creation of a disk group, or to alter the size of a disk group that has been created.

Possible Sector Size/Logical Sector Size Combinations

SECTOR_SIZE/ LOGICAL_SECTOR _SIZE specified at disk creation	512 native disks	512 emulated disks	4K native disks	512 native disks + 512 emulated disks	512 emulated disks + 4K native disks
4K / 4K	Error	4K / 4K	4K / 4K	Error	4K / 4K
NS / 4K	Error	4K / 4K	4K / 4K	Error	4K / 4K
4K / 512	Error	4K / 512	Error	Error	Error
512 / 512	512 / 512	512 / 512	Error	512 / 512	Error
512 / NS	512 / 512	512 / 512	Error	512 / 512	Error
4K / NS	Error	4K / 512	4K / 4K	Error	4K / 4K
NS / 512	512 / 512	512 / 512	Error	512 / 512	Error
NS / NS	512 / 512	512 / 512	4K / 4K	512 / 512	4K / 4K

ORACLE

The table in the slide lists the possible valid combinations of the logical sector size and sector size disk group attributes during disk group creation for disk sizes and types. The first column describes the user-specified values for logical sector size and sector size attributes during disk group creation. NS indicates that no value was specified.

The top row of the table describes the combination of disks used. A 512 native sector size disk has a sector size equal to a logical sector size of 512. The 512 native and 512 emulated headings describe the combination of 512 native disks and 512 emulated disks. The 512 emulated disks have a 4k physical sector size and a 512 logical sector size. The values in each of the table cells represent the final values of the sector sizes of the disk group. Error indicates that the disk group creation failed and an error message is displayed.

Disk Group Sector Size Guidelines

- When migrating from 512 bytes to 4096 bytes, the password files on an ASM disk group need not be migrated or re-created.
- The password files retain their 512 byte block size even after the sector size has been changed to 4096 bytes.
- If the value of the `LOGICAL_SECTOR_SIZE` column is 512, and the value in the `SECTOR_SIZE` column is 4096:
  - The disk group supports 512 sector size emulation
  - I/O operations can be both 512 or 4096 in size
- `SECTOR_SIZE` cannot be smaller than `LOGICAL_SECTOR_SIZE` in an Oracle ASM disk group.
- The sector size attribute setting must be compatible with the physical hardware.

ORACLE

The following guidelines apply to the logical sector size and sector size disk group attributes.

- During the migration process from 512 bytes to 4096 bytes, the password files stored on an Oracle ASM disk group need not be migrated or re-created. The password files are a special case and retain their 512 byte block size even after the sector sizes are changed to 4096 bytes.
- If the value of the `LOGICAL_SECTOR_SIZE` column is 512, and the value of the `SECTOR_SIZE` column is 4096, the disk group supports 512 sector size emulation. This means that I/O operations can be both 512 or 4096 in size. For example, you can have redo log files with a file block size of 512 bytes in the disk group that can be both read and written to, as well as allowing you to create new redo log files with a file block size of 4096 bytes.
- The physical sector size (`SECTOR_SIZE`) cannot be smaller than the logical sector size (`LOGICAL_SECTOR_SIZE`) in an Oracle ASM disk group.
- Not all disks support all possible `SECTOR_SIZE` values. The sector size attribute setting must be compatible with the physical hardware.

Viewing ASM Disk Groups

- In an ASM instance, V$ASM_DISKGROUP displays one row for every ASM disk group discovered by the ASM instance.
- In a database instance, V$ASM_DISKGROUP displays one row for every ASM disk group that is available to the database instance.
- Other ASM dynamic performance views relate to V$ASM_DISKGROUP through the GROUP_NUMBER column.
- The ASMCMD lsdg command provides the same information.
- V$ASM_DISKGROUP_STAT displays performance statistics in the same way that V$ASM_DISKGROUP does, but without performing discovery of new disks.
- The asmcmd lsdg command shows a preformatted view of the data from V$ASM_DISKGROUP.

ORACLE

V$ASM_DISKGROUP contains configuration and status information for ASM disk groups. It is commonly queried by itself to display information about disk groups but is also commonly joined with other ASM views as shown in many of the examples in this lesson. The following example shows a typical query that summarizes space utilization for the ASM disk groups:

```
SQL> SELECT NAME, TYPE, TOTAL_MB, FREE_MB,
  2  REQUIRED_MIRROR_FREE_MB RMFM, USABLE_FILE_MB
  3  FROM V$ASM_DISKGROUP;
NAME   TYPE   TOTAL_MB FREE_MB  RMFM USABLE_FILE_MB
----   ------ -------- ------- ----- --------------
DATA   NORMAL     9998    4248  1449           1399
ACFS   EXTERN     9996    3706     0           3706
FRA    EXTERN     7497    7356     0           7356
```

The asmcmd lsdg command is a preformatted view of the data from V$ASM_DISKGROUP.

```
$ asmcmd lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB
Req_mir_free_MB  Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  NORMAL  N         512   4096  1048576      9998     4248
1449             1399                       0             N  DATA/
MOUNTED  EXTERN  N         512   4096  1048576      7497     7356
0                7356                       0             N  FRA/
```

Note the following when considering the space utilization statistics provided by `V$ASM_DISKGROUP` or `asmcmd lsdg`.

- The `TOTAL_MB` column is the total capacity of the disk group in megabytes, not taking mirroring into account. In the preceding example, both disk groups are `NORMAL` redundancy disk groups; so by default, most files contained in them will be two-way mirrored. In practical terms, this means that the total usable size of both disk groups is approximately half of the number reported.
- The `FREE_MB` column is the total unused capacity of the disk group in megabytes, not taking mirroring into account. In the preceding example, both disk groups are `NORMAL` redundancy disk groups; so in practice, the total free space in both disk groups is approximately half of the number reported.
- The `REQUIRED_MIRROR_FREE_MB` column shows the amount of space that is required to be available in a given disk group in order to restore redundancy after the worst failure that can be tolerated by the disk group.
  In the case of the `DATA` disk group in the preceding example, 977 MB is the size of each of the six failure groups that currently make up that disk group. The worst failure that can be tolerated by this disk group is the loss of one failure group because the loss of any more would mean the loss of any data that was spread across the lost failure groups. So in essence, as long as the `DATA` disk group has 977 MB free, the loss of any one failure group can be tolerated without compromising mirroring.
  In the case of the `FRA` disk group, zero is reported because that disk group consists of only two disks. Because of that, the loss of either disk will compromise mirroring regardless of the amount of space that is free.
- `USABLE_FILE_MB` is computed by subtracting `REQUIRED_MIRROR_FREE_MB` from the total free space in the disk group, and then adjusting the value for mirroring. It is supposed to show the amount of free space that can be safely utilized by taking mirroring into account and yet have the ability to restore redundancy after a disk failure.
  In the case of the `DATA` disk group, the reported negative value shows that mirroring would be compromised by a failure although there is space available in the disk group.
  In the case of the `FRA` disk group, the value has a different meaning. It is already known that the disk group cannot tolerate failure without compromising mirroring. So in this case, the computed value of 630 MB simply refers to the amount of available free space after mirroring is factored in.

Viewing ASM Disk Information

- In an ASM instance, `V$ASM_DISK` displays one row for every disk discovered by the ASM instance, including disks that are not part of any disk group.
- In a database instance, `V$ASM_DISK` displays the same rows as the ASM instance.
- `V$ASM_DISK_STAT` has the same columns but does not discover new disks (may not have the latest information).
- ASMCMD `lsdsk` has several options and two modes.
  - Modes:
    - Connected shows information from dynamic performance views.
    - Nonconnected shows information from disk headers.
  - Options: `-t`, `-k`, and `-p` provide different preformatted views.

An example of a query that retrieves information about disks in a specific disk group is as follows:

```
SQL> SELECT G.NAME DISK_GROUP, D.NAME, D.STATE, D.TOTAL_MB,
  2    D.FREE_MB
  3  FROM V$ASM_DISK D, V$ASM_DISKGROUP G
  4  WHERE D.GROUP_NUMBER = G.GROUP_NUMBER
  5  AND G.NAME = 'DATA';

DISK_GROUP NAME       STATE    TOTAL_MB   FREE_MB
---------- ---------- -----    --------   --------
DATA       DATA_0006  NORMAL    2745       1330
DATA       DATA_0002  NORMAL    2745       1298
DATA       DATA_0001  NORMAL    2745       1305
DATA       DATA_0009  NORMAL    2745       1338
DATA       DATA_0005  NORMAL    2745       1337
DATA       DATA_0007  NORMAL    2745       1345
DATA       DATA_0000  NORMAL    2745       1313
DATA       DATA_0004  NORMAL    2745       1340
DATA       DATA_0008  NORMAL    2745       1334
DATA       DATA_0003  NORMAL    2565       1238
```

Another example that uses V$ASM_DISK to display disks that do not belong to any disk group is as follows:

```
SQL> SELECT PATH, MOUNT_STATUS, HEADER_STATUS, OS_MB
  2  FROM V$ASM_DISK
  3  WHERE GROUP_NUMBER = 0;

PATH              MOUNT_S HEADER_STATU      OS_MB
---------------- ------- ------------ ----------
/dev/asmdisk2p4  CLOSED  CANDIDATE          2353
/dev/asmdisk2p3  CLOSED  CANDIDATE          2627
```

V$ASM_DISK_STAT displays performance statistics in the same way that V$ASM_DISK does, but without performing discovery of new disks. This results in a less expensive operation. However, because discovery is not performed, the output of this view does not include any data about disks that are new to the system.

The columns for V$ASM_DISK_STAT are the same as those for V$ASM_DISK.

The ASMCMD lsdsk command provides a way to see the V$ASM_DISK information in preformatted columns without having to write SQL statements. The ASMCMD lsdsk command can be used in connected mode where it retrieves data from V$ASM_DISK, or nonconnected mode where data is collected from the ASM disk headers. The -g and --discovery options control whether V$ASM_DISK or V$ASM_DISK_STAT is used.

The ASMCMD lsdsk -k command that connects to the ASM instance shows the following:

```
$ asmcmd lsdsk -k
Total_MB  Free_MB  OS_MB  Name       Failgroup  Failgroup_Type  Library  Label
UDID  Product  Redund   Path
    2745     1313   2745  DATA_0000  DATA_0000  REGULAR         System
UNKNOWN  /dev/asmdisk1p1
    2745     1305   2745  DATA_0001  DATA_0001  REGULAR         System
UNKNOWN  /dev/asmdisk1p10
    2745     1298   2745  DATA_0002  DATA_0002  REGULAR         System
UNKNOWN  /dev/asmdisk1p11
    2565     1238   2565  DATA_0003  DATA_0003  REGULAR         System
UNKNOWN  /dev/asmdisk1p12
    2745     1340   2745  DATA_0004  DATA_0004  REGULAR         System
UNKNOWN  /dev/asmdisk1p2
    2745     1337   2745  DATA_0005  DATA_0005  REGULAR         System
UNKNOWN  /dev/asmdisk1p3
    2745     1330   2745  DATA_0006  DATA_0006  REGULAR         System
UNKNOWN  /dev/asmdisk1p4
    2745     1345   2745  DATA_0007  DATA_0007  REGULAR         System
UNKNOWN  /dev/asmdisk1p5
```

Extending an Existing Disk Group

- The ALTER DISKGROUP command enables you to extend an existing disk group by adding disks to it.

```
SQL> ALTER DISKGROUP FRA ADD DISK
'ORCL:SDE5' NAME 'FRA_DISK3','ORCL:SDE6' NAME 'FRA_DISK4';
```

- Add disks to an existing disk group with the ASMCMD CHDG command:

```
ASMCMD> chdg '<chdg name="FRA" power="3"><add><fg name="fg4">
<dsk string="/dev/asmdisk5"/></fg></add></chdg>'
```

- ASMCA and Enterprise Manager also provide interfaces to extend existing disk groups.

You can use the ADD clause of the ALTER DISKGROUP statement to add a disk or a failure group to a disk group. The same syntax that you use to add a disk or failure group with the CREATE DISKGROUP statement can be used with the ALTER DISKGROUP statement.

ASM automatically rebalances the disk group when disks are added. By default, the ALTER DISKGROUP statement returns immediately after the disks are added while the rebalance operation continues to run asynchronously. You can query the V$ASM_OPERATION view to monitor the status of the rebalance operation.

You can optionally use the REBALANCE clause to manually control the rebalance process.

In the POWER clause, you can specify a value from 0 to 1024. A value of 0 disables rebalancing for this statement. A value of 1 causes the rebalance to take place with minimal resources allocated to it, whereas a value of 1024 permits ASM to execute the rebalance as quickly as possible. If you do not specify a value, the POWER clause defaults to the value of the ASM_POWER_LIMIT initialization parameter.

You can specify WAIT or NOWAIT to determine whether the ALTER DISKGROUP statement waits for the rebalance to conclude before returning or not. NOWAIT is the default.

You can also extend a disk group by using the Add Disks button in the ASM Disk Groups window of the Database Configuration Assistant (DBCA). Enterprise Manager provides an Add button in each specific Disk Group window.

Dropping Disks from an Existing Disk Group

- The ALTER DISKGROUP command enables you to remove disks or failure groups from an existing disk group:

```
SQL> ALTER DISKGROUP FRA DROP DISK FRA_DISK1, FRA_DISK4;
```

- Use ASMCMD to remove disks or failure groups:

```
$ cat chg_fra.xml
<chdg name="FRA" power="3">
<drop>
<fg name="fg4"></fg>
<dsk name="/dev/asmdisk5"/>
</drop>
</chdg>

ASMCMD> chdg chg_fra.xml
```

ORACLE

To drop disks from a disk group, use the DROP DISK clause of the ALTER DISKGROUP statement. A dropped disk can be reused by adding it to a disk group or by using it in the creation of a new disk group. A dropped disk can be added back to the disk group it was removed from. When a disk is dropped, the disk group is rebalanced by moving all the file extents from the dropped disk to other disks in the disk group. A drop disk operation fails if not enough space is available on the other disks. Data will not be lost by dropping a disk.

By default, the ALTER DISKGROUP...DROP DISK statement returns before the drop and rebalance operations are complete. Do not reuse, remove, or disconnect the dropped disk until the HEADER_STATUS column for this disk in the V$ASM_DISK view changes to FORMER. You can query the V$ASM_OPERATION view to determine the amount of time remaining for the drop and rebalance operations to complete.

If you specify the FORCE clause for the drop operation, the disk is dropped even if ASM cannot read from or write to the disk. You cannot use the FORCE flag when dropping a disk from an external redundancy disk group. A DROP FORCE operation leaves data at reduced redundancy for as long as it takes for the subsequent rebalance operation to complete. Take great care because this increases your exposure to data loss if there is subsequent disk failure during rebalancing. The REBALANCE clause works in the same way as when extending a disk group. You can also use Enterprise Manager, ASMCA, or ASMCMD interfaces to drop disks from a disk group.

REBALANCE POWER 0

- You can effectively disable rebalancing when adding disks to or removing disks from a disk group by specifying `REBALANCE POWER 0` in the `ALTER DISKGROUP` statement.
- When adding, the disks become immediately available to the disk group; however, existing data is not moved to the new disks.
  - New data may be written to the new disks.
- When removing, the statement executes and the disks are marked with a status of `DROPPING`. However, because the operation to move data to another disk is effectively disabled, the disk remains in this state indefinitely until another operation causes the disk group to be rebalanced.

ORACLE

When adding disks to or removing disks from a disk group, you can effectively disable rebalancing by specifying `REBALANCE POWER 0` in the `ALTER DISKGROUP` statement.

When adding disks to a disk group in conjunction with `REBALANCE POWER 0`, the disks become immediately available to the disk group. However, existing data is not moved to the new disks. New data may be written to the new disks. This is useful in situations where you need to add disks quickly to a running system and you want to defer the rebalance operation until a later time such as a scheduled maintenance period.

When removing disks from a disk group in conjunction with `REBALANCE POWER 0`, the statement executes and the disks are marked with a status of `DROPPING`. However, because the operation to move data to another disk is effectively disabled, the disk remains in this state indefinitely until another add or drop operation causes the disk group to be rebalanced or the disk group is manually rebalanced by using the `ALTER DISKGROUP <diskgroup_name> REBALANCE` statement.

V$ASM_OPERATION

- In an ASM instance, V$ASM_OPERATION displays one row for every active, long-running operation that is executing in the ASM instance.
- In a database instance, V$ASM_OPERATION displays no rows.
- The ASMCMD lsop command shows ASM operations:

```
ASMCMD> lsop -G DATA
Group_Name Pass       State Power EST_WORK EST_RATE EST_TIME
DATA       RESYNC     DONE  10    0        0        0
DATA       COMPACT    WAIT  10    0        0        0
DATA       REBALANCE  RUN   10    136      229      0
```

ORACLE

V$ASM_OPERATION provides information about the long-running operations performed by ASM. A long-running operation is one of the following:

- A rebalance operation that results from adding disks to or removing disks from a disk group
- A mirror resync operation that results from a disk being brought back online after an outage

The following example shows a typical query that lists all current operations:

```
SQL> SELECT G.NAME DISK_GROUP, O.*
  2  FROM V$ASM_OPERATION O, V$ASM_DISKGROUP G
  3  WHERE O.GROUP_NUMBER = G.GROUP_NUMBER;

DISK_GROUP GROUP_NUMBER OPERATION STATE POWER ACTUAL
---------- ------------ --------- ----- ----- ------
SOFAR EST_WORK EST_RATE EST_MINUTES ERROR_CODE
----- -------- -------- ----------- ----------
DATA                  1 REBAL     RUN       2      2
  644     1237      565           1
```

In this example, a rebalance operation is currently running. The operation is estimated to require the movement of 1,237 allocation units. So far, 644 allocation units have been moved at a rate of 565 per minute. It is estimated that the operation will complete in 1 minute. No errors have been reported.

Adding and Dropping in the Same Command

You can add and drop disks at the same time by using:

- The `ASMCMD CHDG` command:

```
$ cat chg_data.xml
<chdg name="data" power="4">
  <drop>
    <fg name="fg1"></fg>
    <dsk name="data_0001"/>
  </drop>
 <add>
   <fg name="fg2">
   <dsk string="/dev/disk5"/>
   </fg>
  </add>
</chdg>
ASMCMD> chd_data.xml
```

- The `ALTER DISKGROUP` command:

```
SQL> ALTER DISKGROUP FRA
ADD DISK 'ORCL:SDE7' NAME 'FRA_DISK5' SIZE 977 M ,
         'ORCL:SDE8' NAME 'FRA_DISK6' SIZE 977 M
DROP DISK FRA_DISK1, FRA_DISK2;
```


It is possible to combine both the add and drop operations in the same `ALTER DISKGROUP` statement. This approach is recommended in preference to separate operations wherever possible. Adding and dropping in the same command has the benefit of rebalancing data extents once, and can provide greater assurance that there is enough space for the rebalance operation to succeed.

Adding disks to a disk group and dropping disks from a disk group, whether in the same command or not, is an effective way of migrating ASM from an existing storage platform to a new one. For example, when a new disk subsystem is configured for discovery by ASM, it is possible to migrate your data from its current storage to the new storage without down time by using a single command for each disk group.

Undropping Disks in Disk Groups

Use `ALTER DISKGROUP ... UNDROP DISKS` to cancel all pending disk removal operations within disk groups.

```
ALTER DISKGROUP
  { diskgroup_name [, diskgroup_name ] ...
  | ALL
  }
  UNDROP DISKS
;
```

Examples:

```
ALTER DISKGROUP DATA UNDROP DISKS;
```

```
ALTER DISKGROUP DATA2, DATA3 UNDROP DISKS;
```

```
ALTER DISKGROUP ALL UNDROP DISKS;
```


The `UNDROP DISKS` clause of the `ALTER DISKGROUP` statement enables you to cancel all pending disk removal operations within disk groups. Pending disk removal operations include those that are currently in progress.

This statement cannot be used to restore disks that are being dropped as the result of a `DROP DISKGROUP` statement, or for disks that are being dropped by using the `FORCE` clause.

If a drop disk operation has already completed, this statement cannot be used to restore it. In this case, you can simply add the dropped disk back to the disk group to achieve the same outcome.

Replacing Disks in Disk Groups

- A disk or multiple disks in a disk group can be replaced, rather than being dropped and added back.
- A single replace operation is more efficient than dropping and adding disks.
- A `REPLACE` operation is especially useful when disks are missing or damaged.
- To replace the `diskC7` disk with another disk identified by the `/devices/diskC18`:

```
SQL> ALTER DISKGROUP data2 REPLACE DISK diskC7 WITH '/devices/diskC18' POWER 3;
```

  - The power option cannot be set to 0.

A disk or multiple disks in a disk group can be replaced, rather than being dropped and added back. The single replace operation is more efficient than dropping and adding disks. This operation is especially useful when disks are missing or damaged.

For example, you can issue the following statement to replace the `diskC7` disk with another disk that is identified by the `/devices/diskC18` path.

```
SQL> ALTER DISKGROUP data2 REPLACE DISK diskC7 WITH '/devices/diskC18' POWER 3;
```

The power option cannot be set to 0. The `ALTER DISKGROUP` SQL statement with the `REPLACE` clause includes a `WAIT` or `NOWAIT` option, plus the `FORCE` option.

Renaming Disk Groups

- The `renamedg` utility enables you to change the name of a cloned disk group.
- `renamedg` renames a disk group by using a two-step process:
  1. *Phase one:* This phase generates a configuration file to be used in phase two.
  2. *Phase two:* This phase uses the configuration file to perform the renaming of the disk group.
- To rename the `fra1` disk group to `fra2` by using a disk string to locate the disks:

```
$ renamedg dgname=fra1 newdgname=fra2 asm_diskstring='/devices/disk*'
  verbose=true
```

- To create a config file during the completion of phase one:

```
$ renamedg phase=one dgname=fra1 newdgname=fra2
asm_diskstring='/devices/disk*' config=/tmp/fra2.conf verbose=true
```

ORACLE

The `renamedg` tool enables you to change the name of a cloned disk group. The disk group must be dismounted on all nodes in the cluster before running `renamedg` on the disk group.

The `renamedg` utility renames a disk group by using a two-step process:

1. **Phase one:** This phase generates a configuration file to be used in phase two.
2. **Phase two:** This phase uses the configuration file to perform the renaming of the disk group.

The syntax is:

```
renamedg {-help | help=true}
renamedg
[phase={ one|two |both } ] dgname=diskgroup newdgname=newdiskgroup
[config=configfile][ asm_diskstring=discoverystring, discoverystring ... ] [
clean={true|false} ] [ check={true|false} ]                    [
confirm={true|false}] [ verbose={ true|false} ]
[ keep_voting_files={true|false}]
```

- **`phase`:** Specifies the phase to be run. The values that are allowed are one, two, or both. This argument is optional. The default is both.
- **`dgname`:** Specifies the name of the disk group that to be renamed.
- **`newdgname`:** Specifies the new name for the disk group
- **`config`:** Specifies the path to the configuration file to be generated during phase one or specifies the path to the configuration file to be used during phase two
- **`asm_diskstring`:** Specifies the Oracle ASM discovery strings
- **`clean`:** Specifies whether to clean errors that are otherwise ignored. The default is true.
- **`check`:** Specifies a Boolean value that is used in the second phase. If true, the tool prints the list of changes that are to be made to the disks. No writes are issued. It is an optional parameter that defaults to false.

- **`confirm`:** Specifies a Boolean value that is used in the second phase. If false, the tool prints the changes that are to be made and seeks confirmation before actually making the changes.
- **`keep_voting_files`:** Specifies whether voting files are kept in the renamed disk group. The default is false, which deletes the voting files from the renamed disk group.

In the slide, the first example renames the `fra1` disk group to `fra2` by using a disk string to locate the disks and the verbose option is enabled. The second example creates a configuration file only during the completion of phase one of the `renamedg` operation.

To run phase two of the `renamedg` operation by using the configuration file that is generated from the phase one execution of `renamedg`:

```
$ renamedg phase=two dgname=fra1 newdgname=fra2 config=/tmp/fra2.conf
verbose=true
```

After renaming a disk group, you can rename the disks in the disk group to match the new disk group name. For example:

```
SQL> ALTER DISKGROUP fra2 RENAME DISKS ALL;
```

Note: `renamedg` does not update resources, nor does `renamedg` update any file references within the database. Because of this behavior, the original disk group resource is not automatically deleted after the completion of phase two.

Renaming Disks in Disk Groups

- You can rename a disk in a disk group with the ALTER DISKGROUP RENAME DISK SQL statement.
- In one statement, you can rename one or multiple disks, or rename all the disks in a disk group.
- To rename disk FRA1_0001 to FRA2_0001:

```
SQL> ALTER DISKGROUP fra2 MOUNT RESTRICTED;
SQL> ALTER DISKGROUP fra2 RENAME DISK 'FRA1_0001' TO 'FRA2_0001', 'FRA1_0002'
TO 'FRA2_0002';
```

- You can use RENAME DISK only when the disk group that contains the disk is in the MOUNT RESTRICTED state.
- The RENAME operation can be run after renamedg to change the names of the disks in the renamed disk group.

You can rename a disk in a disk group with the ALTER DISKGROUP RENAME DISK SQL statement. In one SQL statement, you can rename one or multiple disks, or rename all the disks in a disk group by using the RENAME DISKS ALL clause.

For example, you can rename disks as follows:

The ALTER DISKGROUP *diskgroupname* RENAME DISKS ALL statement can be run after the renamedg utility to change the names of the disks in the renamed disk group.

When you run the ALTER DISKGROUP *diskgroupname* RENAME DISKS ALL statement, any disk name that is not in the format *diskgroupname_number* is renamed to that format. Disk names that are already in the *diskgroupname_number* format are not changed.

You can use the RENAME DISK operation only when the disk group that contains the disk is in the MOUNT RESTRICTED state. If any disks in the disk group are offline, the RENAME operation fails. If the new disk name exists, the RENAME operation fails. You must have SYSASM privileges to rename a disk.

Resizing Disks in Disk Groups

- The `RESIZE` clause of `ALTER DISKGROUP` enables you to perform the following operations:
  - Resize all the disks in a disk group.
  - Resize specific disks.
  - Resize all the disks in a specified failure group.
- If the size of the disk is increasing, the new space is immediately available for allocation.
- If the size is decreasing, rebalancing must relocate extents beyond the new size limit to available space below the limit.
- To resize all the disks in the `failgrp1` failure group of disk group `data1`:

```
SQL> ALTER DISKGROUP data1 RESIZE DISKS IN FAILGROUP failgrp1 SIZE 100G;
```


The `RESIZE` clause of `ALTER DISKGROUP` enables you to perform the following operations:

- Resize all the disks in the disk group.
- Resize specific disks.
- Resize all the disks in a specified failure group.

If you do not specify a new size in the `SIZE` clause, Oracle ASM uses the size of the disk as returned by the operating system. The new size is written to the Oracle ASM disk header and if the size of the disk is increasing, the new space is immediately available for allocation. If the size is decreasing, rebalancing must relocate file extents beyond the new size limit to available space below the limit. If the rebalance operation can successfully relocate all extents, the new size is made permanent; otherwise, the rebalance operation fails.

The example in the slide resizes all the disks in the `failgrp1` failure group of the `data1` disk group. If the new size is greater than disk capacity, the statement fails.

Mounting and Dismounting Disk Groups

- The SQL ALTER DISKGROUP command enables you to manually mount and dismount disk groups:

```
SQL> ALTER DISKGROUP data1 RESIZE DISKS IN FAILGROUP failgrp1 SIZE 100G;
```

```
SQL> ALTER DISKGROUP DATA2, DATA3 MOUNT;
```

```
SQL> ALTER DISKGROUP ALL DISMOUNT;
```

- Mounting and unmounting disk groups by using ASMCMD:

```
ASMCMD> mount -f DATA
```

```
ASMCMD> mount --restrict DATA2
```

```
ASMCMD> umount -f DATA2
```


Disk groups must be mounted for database instances to access the files they contain. To mount or dismount disk groups, use the MOUNT or DISMOUNT clauses of the ALTER DISKGROUP statement. You can mount or dismount disk groups by name, or specify ALL. Specify ALL MOUNT to mount all the disk groups specified in the ASM_DISKGROUPS initialization parameter. Specify ALL DISMOUNT to dismount all mounted disk groups.

If a disk group is mounted by using the RESTRICTED option, no other ASM instance in the same cluster can mount that disk group and the disk group is not usable by any database instances. RESTRICTED mode can be used to improve rebalancing performance on a cluster by eliminating messaging and coordination with other ASM instances and database instances.

For normal and high redundancy disk groups, use the FORCE option of the MOUNT clause to mount a disk group if there are sufficient ASM disks available. The mount succeeds if ASM finds at least one complete set of extents in a disk group. MOUNT FORCE is used in situations where one or more disks are not available. When MOUNT FORCE is used, ASM flags the unavailable disks as offline and drops the disks after DISK_REPAIR_TIME expires. If you try to dismount a disk group that contains open files, the statement fails unless the FORCE clause is included. If you perform a DISMOUNT FORCE, the files in the disk group become inaccessible to your databases, and any operations involving those files fail. ASMCMD can be used to mount and mount disk groups as shown in the slide. Use -a to mount all disks. The -f option can be used to force the mount operation if enough disks are available.

Viewing Connected Clients

- In an ASM instance, V$ASM_CLIENT displays one row for each open ASM disk group used by each client database instance.
- In a database instance, V$ASM_CLIENT displays one row for each open ASM disk group used by the database instance.
- Connect to an ASM instance with the asmcmd lsct command

```
$ asmcmd lsct -g data
Inst_ID DB_Name Status Software_Version Compatible_version Inst_Name Dsk_Grp
      1 +ASM    CONNECTED    12.1.0.1.0        12.1.0.1.0 +ASM1     DATA
      2 +ASM    CONNECTED    12.1.0.1.0        12.1.0.1.0 +ASM2     DATA
      3 +ASM    CONNECTED    12.1.0.1.0        12.1.0.1.0 +ASM3     DATA
      2 orcl    CONNECTED    12.1.0.1.0        12.1.0.0.0 orcl_1    DATA
```


A typical example of a query that retrieves information about the databases that are current clients of a particular disk group by using V$ASM_CLIENT from the ASM instance is as follows:

```
SQL> SELECT G.NAME DISK_GROUP, C.*
  2  FROM V$ASM_CLIENT C, V$ASM_DISKGROUP G
  3  WHERE C.GROUP_NUMBER = G.GROUP_NUMBER
  4  AND G.NAME = 'DATA';
                                                             SOFTWARE
   COMPATIBLE
DISK_GROUP GROUP# INSTANCE_NAME DB_NAME  STATUS    VERSION    VERSION
---------- ------ ------------- -------- --------- ---------- ----------
DATA            1 orcl_3        orcl     CONNECTED 12.1.0.1.0 12.1.0.1.0
DATA            1 +ASM1         +ASM     CONNECTED 12.1.0.1.0 12.1.0.1.0
```

The asmcmd lsct command provides similar information.

Dropping Disk Groups

- The SQL DROP DISKGROUP statement enables you to delete a disk group and, optionally, all its files:

```
SQL> DROP DISKGROUP DATA2;
```

```
SQL> DROP DISKGROUP DATA2 INCLUDING CONTENTS;
```

```
SQL> DROP DISKGROUP DATA3 FORCE INCLUDING CONTENTS;
```

- ASMCMD can be used to drop a disk group, including its contents, if required:

```
ASMCMD> dropdg DATA2;
```

```
ASMCMD> dropdg –r DATA2;
```

```
ASMCMD> dropdg –r –f DATA3;
```


The DROP DISKGROUP statement enables you to delete an ASM disk group and optionally, all its files. You can specify the INCLUDING CONTENTS clause if you also want to delete any files that might be contained in the disk group. The default is EXCLUDING CONTENTS, which prevents you from dropping the disk group if it has any contents.

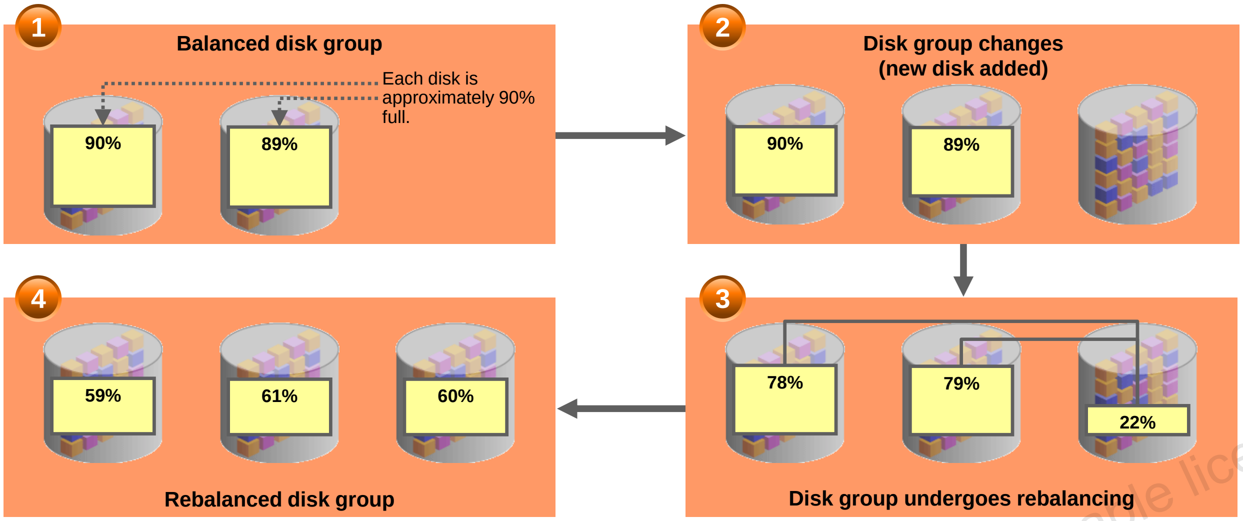
For the DROP DISKGROUP statement to succeed, the disk group must be mounted only by the current ASM instance and have none of the disk group files open.

When you drop a disk group, ASM dismounts the disk group and removes the disk group name from the ASM_DISKGROUPS initialization parameter if an SPFILE file is being used. If a text initialization parameter file (PFILE) is being used, you must manually adjust the ASM_DISKGROUPS initialization parameter to make sure that ASM does not attempt to mount the dropped disk group the next time the instance starts.

If you cannot mount a disk group but need to drop it, you can use the FORCE INCLUDING CONTENTS option. This enables you to remove the headers on disks that belong to a disk group that cannot be mounted by any ASM instances. When you use the FORCE INCLUDING CONTENTS option, the ASM instance does not attempt to verify that the disk group is being used by another ASM instance.

ASMCMD can also be used to drop disk groups. Use the –r option to drop a disk group and its contents. Use the –f option to force the operation.

ASM Disk Group Rebalance: Review



An ASM disk group is balanced when there is even distribution of data across all available disks. In normal course of operations, ASM maintains balanced disk groups. A disk group can become unbalanced for various reasons, most commonly when disks are either added to or removed from the disk group. To restore balance, a rebalance operation is required, and it involves moving ASM extents from the most filled disks to the least filled disks. Enhancements have been made to the ASM disk group rebalance operation in Oracle Database 12*c* ASM. The following pages introduce those enhancements.

Priority Ordered Rebalance

- Sometimes, rebalance operations are required to restore redundancy. For example, a disk fails and no replacement is available.
- In earlier versions:
  - The rebalance occurs in file-number order.
  - A secondary failure could result in the loss of a critical file.
- With Oracle Database 12*c* ASM:
  - Critical files, such as control files and log files, are restored before data files.
  - Secondary failure is less likely to result in critical file loss.

ORACLE

In some situations, rebalance operations are required to restore data redundancy within disk groups that use `NORMAL` or `HIGH` ASM redundancy. For example, if a disk fails and no replacement is available, a rebalance is required to redistribute data across the remaining disks and to restore redundancy.

With Oracle Database 12*c* ASM, priority ordered rebalance is implemented. This capability concentrates on quickly restoring the redundancy of critical files, such as control files and online redo log files, to ensure that they are protected in case a secondary failure occurs soon afterwards.

Tuning Rebalance Operations

- If the POWER clause is not specified, the rebalance power defaults to the value of ASM_POWER_LIMIT.
- The value of ASM_POWER_LIMIT can be set dynamically.
- The EXPLAIN WORK statement gauges the amount of work for a rebalance. View the results in V$ASM_ESTIMATE.

```
SQL> EXPLAIN WORK FOR ALTER DISKGROUP data DROP DISK data_0000;
Explained.
SQL> SELECT est_work FROM V$ASM_ESTIMATE;
EST_WORK
--------
2573
SQL> EXPLAIN WORK SET STATEMENT_ID='online' FOR ALTER DISKGROUP data
ONLINE disk data_000;
Explained.
SQL> SELECT est_work FROM V$ASM_ESTIMATE WHERE STATEMENT_ID='online';
EST_WORK
--------
421
```


Earlier ASM releases provided work estimates for ASM rebalance operations. However, the accuracy of these estimates was highly variable. Oracle Database 12*c* ASM provides more accurate rebalance work estimates.

If the POWER clause is not specified in an ALTER DISKGROUP statement, or when rebalance is implicitly run by adding or dropping a disk, the rebalance power defaults to the value of the ASM_POWER_LIMIT initialization parameter.

You can adjust the value of this parameter dynamically. The range of values for the POWER clause is the same as that for the ASM_POWER_LIMIT initialization parameter. The higher the power limit, the more quickly a rebalance operation completes. Rebalancing takes longer with lower power values, but consumes fewer processing and I/O resources, which are shared by other applications, such as the database.

ASM tries to keep a rebalance I/O for each unit of power. Each I/O requires PGA memory for the extent that is involved in the relocation. The default value of 1 minimizes disruption to other applications. The appropriate value is dependent on your hardware configuration, performance requirements, and availability requirements. Oracle ASM always adjusts the power to fit available memory.

When the COMPATIBLE.ASM disk group is set to 11.2.0.2 or later, the rebalance operation may be run as one process by using asynchronous I/O. You can check the ASM alert log for details on the rebalance process. If a rebalance is in progress because a disk is automatically or manually dropped, increasing the power of the rebalance shortens the time frame during which redundant copies of that data on the dropped disk are reconstructed on other disks.

You can also affect rebalance behavior with the CONTENT.CHECK and THIN_PROVISIONED disk group attributes. The EXPLAIN WORK SQL statement determines the amount of work for a rebalance operation and the resulting calculations are displayed in the V$ASM_ESTIMATE view as shown in the slide. The EST_WORK column provides an estimate of the number of allocation units that must be moved by the rebalance operation to complete.

The PASS column of V$ASM_OPERATION is updated for the resync and rebalance operations. The contents of the column can be RESYNC, REBALANCE, or COMPACT. For example, the following SQL query shows values in the PASS column during a rebalance operation.

```
SQL> SELECT GROUP_NUMBER, PASS, STATE FROM V$ASM_OPERATION;

GROUP_NUMBER PASS        STAT

------------ ---------   ----

           2 RESYNC      WAIT

           2 REBALANCE   WAIT

           2 COMPACT     WAIT
```

Quiz

If you create a disk group and do not specify the `REDUNDANCY` clause, the default setting will be:

a. `EXTERNAL`

b. `NORMAL`

c. `HIGH`

d. The default varies depending on the number of failure groups that are defined.

Answer: b

However, if only one failure group is specified, the `CREATE DISKGROUP` statement fails. In any event, it is recommended that the `REDUNDANCY` clause should be specified explicitly because the setting cannot be changed after the disk group has been created.

Quiz

Adding a disk to a disk group will always cause a rebalance operation to occur.

a. True

b. False

Answer: b

If the `ASM_POWER_LIMIT` initialization parameter is set to zero or `REBALANCE POWER 0` is specified in the `ALTER DISKGROUP` statement, rebalancing is disabled.

Summary

In this lesson, you should have learned how to:

- Create and delete ASM disk groups
- Set the attributes of an existing ASM disk group
- Perform ongoing maintenance tasks on ASM disk groups

ORACLE

Practice 4 Overview: Administering ASM Disk Groups

This practice covers the following topics:

- Configuring disk groups
- Adding and removing disks
- Controlling rebalance operations
- Renaming disk groups

5

Administering ASM Disk Groups Part II

Objectives

After completing this lesson, you should be able to:

- Explain capacity management for external, normal, and high redundancy disk groups.
- Describe the concept of partner status tables and partner disks.
- Explain the functionality of ASM Fast Mirror Resync.
- View important ASM disk statistics.
- Explain key performance and scalability considerations for ASM disk groups

ORACLE

Managing Capacity in Disk Groups

- In a normal or high redundancy disk group, there must be enough capacity to manage re-creation of data that is lost after failure of one or two failure groups.
  - If there is not enough space, some files might end up with reduced redundancy.
- Reduced redundancy means one or more extents in the file are not mirrored at the expected level.
- If the disk group redundancy is:
  - Normal: There should be enough free space to tolerate the loss of all disks in one failure group.
  - High: There should be enough free space to cope with the loss of all disks in two failure groups.

When ASM provides redundancy, such as when you create a disk group with `normal` or `high` redundancy, there must be enough capacity in each disk group to manage re-creation of data that is lost after failure of one or two failure groups. After one or more disks fail, the process of restoring redundancy for all data requires space from the surviving disks in the disk group. If there is not enough space, some files might end up with reduced redundancy.

Reduced redundancy means one or more extents in the file are not mirrored at the expected level. For example, a reduced redundancy file in a high redundancy disk group has at least one extent with two or fewer total copies of the extent instead of three. For unprotected files, data extents could be completely missing. Other causes of reduced redundancy files are disks running out of space or an insufficient number of failure groups. To ensure that you have sufficient space to restore full redundancy for all disk group data after the failure of one or more disks, consider the following guidelines:

- **Normal redundancy disk group:** It is best to have enough free space in your disk group to tolerate the loss of all disks in one failure group. The amount of free space should be equivalent to the size of the largest failure group.
- **High redundancy disk group:** It is best to have enough free space to cope with the loss of all disks in two failure groups. The amount of free space should be equivalent to the sum of the sizes of the two largest failure groups.

Managing Capacity in Disk Groups

- The information in V$ASM_DISKGROUP can help you to manage disk group capacity.

```
SQL> SELECT name, type, total_mb, free_mb,
required_mirror_free_mb, usable_file_mb FROM V$ASM_DISKGROUP;

NAME   TYPE   OTAL_MB  FREE_MB REQ_MIRROR_FREE_MB USABLE_FILE_MB
---- ------   ------- -------- ------------------ --------------
DATA  NORMAL    27609    18643               2761           7941
FRA   EXTERN     8282     7801                  0           7801
```

- The asmcmd lsdg command provides similar information:

```
ASMCMD> lsdg
State   Type    ... Total_MB Free_MB Req_mir_free_MB Usable_file_MB ... Name
MOUNTED NORMAL ...    27609   18643            2761           7941 ... DATA/
MOUNTED EXTERN ...     8282    7801               0           7801 ... FRA/
```

ORACLE

The V$ASM_DISKGROUP view contains the following columns that contain information to help you manage capacity:

- REQUIRED_MIRROR_FREE_MB indicates the amount of space that must be available in a disk group to restore full redundancy after the worst failure that can be tolerated by the disk group without adding additional storage. This requirement ensures that there are sufficient failure groups to restore redundancy. Also, worst failure refers to a permanent failure where the disks must be dropped, not the case where disks go offline, and then come back online.
- USABLE_FILE_MB indicates the amount of free space, adjusted for mirroring, that is available for new files to restore redundancy after a disk failure. USABLE_FILE_MB is computed by subtracting REQUIRED_MIRROR_FREE_MB from the total free space in the disk group, and then adjusting the value for mirroring. For example, in a normal redundancy disk group where by default the mirrored files use disk space that is equal to twice their size, if 4 GB of actual usable file space remains, USABLE_FILE_MB equals approximately 2 GB. You can then add a file that is up to 2 GB.
- TOTAL_MB is the total usable capacity of a disk group in megabytes. Calculations for data in this column take the disk header overhead into consideration. The disk header overhead depends on the number of ASM disks and files. This value is typically about 1% of the total raw storage capacity. For example, if the total LUN capacity provisioned for ASM is 100 GB, the value in the TOTAL_MB column would be about 99 GB.
- FREE_MB is the unused capacity of the disk group in megabytes, without considering any data imbalance. There may be situations where the value in the FREE_MB column shows unused capacity but because one ASM disk is full, database writes fail due to the imbalance in the disk group. Ensure that you initiate a manual rebalance to force even data distribution, which results in accurate presentation of values in the FREE_MB column.

  With fine-grain striping using 128 KB, storage is pre-allocated to be eight times the AU size. The data file size may appear slightly larger on ASM than on a local file system because of the pre-allocation.

  When you use ASM normal or high redundancy, disk space utilization becomes more complex to measure because it depends on several variables.

Capacity Calculation: Example

- Consider the `DATA` disk group that uses normal redundancy and consists of six 1 GB disks, each in its own failure group:

```
SQL> SELECT name, type, total_mb, free_mb, required_mirror_free_mb,
usable_file_mb FROM V$ASM_DISKGROUP;
NAME TYPE    TOTAL_MB    FREE_MB    REQUIRED_MIRROR_FREE_MB USABLE_FILE_MB
---- ------ ---------- ---------- ----------------------- --------------
DATA NORMAL       6144       3768                    1024           1372
```

- `REQUIRED_MIRROR_FREE_MB` reports that 1 GB must be available to restore redundancy after one or more disks fail.
- `USABLE_FILE_MB` is calculated as:

```
(FREE_MB - REQUIRED_MIRROR_FREE_MB) / 2 = USABLE_FILE_MB

(3768 MB - 1024 MB) / 2 =

2744 / 2 = 1372 → USABLE_FILE_MB
```


When you use ASM normal or high redundancy, disk space utilization becomes more complex to measure because it depends on several variables. The results from the query in the slide show capacity metrics for a normal redundancy disk group that consists of six 1 GB disks, each in its own failure group.

The `REQUIRED_MIRROR_FREE_MB` column shows that 1 GB of extra capacity must be available to restore full redundancy after one or more disks fail. The first three numeric columns in the query results are raw numbers. That is, they do not take redundancy into account. Only the last column is adjusted for normal redundancy. In the query output example for the `DATA` disk group, the calculation is as follows:

```
(FREE_MB - REQUIRED_MIRROR_FREE_MB) / 2 = USABLE_FILE_MB
(3768 - 1024) / 2 = 2744 / 2 = 1372
```

For high redundancy disk groups, `USEABLE_FILE_MB` is calculated as `(FREE_MB - REQUIRED_MIRROR_FREE_MB) / 3`.

Due to the relationship between `FREE_MB`, `REQUIRED_MIRROR_FREE_MB`, and `USABLE_FILE_MB`, `USABLE_FILE_MB` can become negative. Although this is not necessarily a critical situation, it does mean that:

- Depending on the value of `FREE_MB`, you may not be able to create new files
- The next failure might result in files with reduced redundancy

If `USABLE_FILE_MB` becomes negative, it is strongly recommended that you add more space to the disk group as soon as possible.

Normal redundancy disk groups that consist of at least three failure groups require `REQUIRED_MIRROR_FREE_MB` to be the size of a single failure group. If there are only two failure groups, `REQUIRED_MIRROR_FREE_MB` would be the size of a single disk.

High redundancy disk groups that consist of at least four failure groups require an amount of free space that is equal to the size of two failure groups. If a high redundancy disk group has only three failure groups, the amount of free space required would be the size of two disks.

Partner Status Table

- The Partner Status Table (PST) stores information about the ASM disks that are contained in a disk group.
- This information includes:
  - Disk number
  - Disk status (online or offline)
  - Partner disk number
  - Failure group information
  - Heartbeat status
- Allocation Unit 1 on each disk is reserved for the PST.
- Not all disks store a copy of the PST.

The ASM Partner Status Table (PST) stores status information about the ASM disks that are contained in a disk group. This information includes the disk number, disk status (online or offline), partner disk number, failure group information, and heartbeat status. Allocation Unit 1 on each disk is reserved for the PST but not all disks store a copy of the PST. An external redundancy disk group will have only a single PST. A normal redundancy disk group will have no fewer than two copies of the PST. If there are three or more failure groups, there will be three copies of the PST. A high redundancy disk group will have a minimum of three PSTs. If there are four failure groups, there will be four copies of the PST. If the high redundancy disk group has five or more failure groups, there will be five copies of the PST.

Consider the information logged by the creation of a disk group called `DATA2` by using normal redundancy and three failure groups:

```
SQL> CREATE DISKGROUP DATA2 NORMAL REDUNDANCY  FAILGROUP FGRP1 DISK '/dev/asmdisk2p10'
SIZE 650M ,'/dev/asmdisk2p11' SIZE 650M  FAILGROUP FGRP2 DISK '/dev/asmdisk2p3' SIZE
650M ,'/dev/asmdisk2p4' SIZE 650M  FAILGROUP FGRP3 DISK '/dev/asmdisk2p5' SIZE 650M
,'/dev/asmdisk2p6' SIZE 650M
...
NOTE: Assigning number (3,0) to disk (/dev/asmdisk2p10)
NOTE: Assigning number (3,1) to disk (/dev/asmdisk2p11)
NOTE: Assigning number (3,2) to disk (/dev/asmdisk2p3)
NOTE: Assigning number (3,3) to disk (/dev/asmdisk2p4)
NOTE: Assigning number (3,4) to disk (/dev/asmdisk2p5)
```

```
NOTE: Assigning number (3,5) to disk (/dev/asmdisk2p6)
...
NOTE: Disk 0 in group 3 is assigned fgnum=1
NOTE: Disk 1 in group 3 is assigned fgnum=1
NOTE: Disk 2 in group 3 is assigned fgnum=2
NOTE: Disk 3 in group 3 is assigned fgnum=2
NOTE: Disk 4 in group 3 is assigned fgnum=3
NOTE: Disk 5 in group 3 is assigned fgnum=3
...
NOTE: initiating PST update: grp = 3
Wed May 20 10:19:18 2015
GMON updating group 3 at 12 for pid 7, osid 6038
NOTE: group DATA2: initial PST location: disk 0000 (PST copy 0)
NOTE: group DATA2: initial PST location: disk 0002 (PST copy 1)
NOTE: group DATA2: initial PST location: disk 0004 (PST copy 2)
```

You can see that a copy of the PST is kept on a disk in all three failure groups. Use `kfed` to look at disk 0 (`/dev/asmdisk2p10`), which should have a copy of the PST as indicated in the preceding section:

```
[grid@host01 ~]$ kfed read /dev/asmdisk2p10 -aunum=1|more
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           17 ; 0x002: KFBTYP_PST_META
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                     256 ; 0x004: blk=256
kfbh.block.obj:              2147483648 ; 0x008: disk=0
```

Using `kfed` to look at disk 1 (`/dev/asmdisk2p11`), which should not have a copy of the PST, you see:

```
[grid@host01 ~]$ kfed read /dev/asmdisk2p11 -aunum=1|more
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           13 ; 0x002: KFBTYP_PST_NONE
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:              2147483648 ; 0x004: blk=0 (indirect)
kfbh.block.obj:              2147483649 ; 0x008: disk=1
```

The PST must be available before a disk group can be mounted. When a disk group is mounted, the GMON process for the instance that is making the request reads all the disks in the disk group, verifying the copies of the PST. If it finds enough PSTs to comprise a quorum, the disk group is mounted.

Configure a minimum of three failure groups for normal redundancy disk groups and five failure groups for high redundancy disk groups to maintain the necessary number of copies of the PST to ensure robustness with respect to storage hardware failures.

Partner Disks

- ASM provides data redundancy by placing copies of a disk's extents on partner disks that are located in different failure groups.
- Each ASM disk in a normal or high redundancy disk group can have up to eight partner disks.
- The `X$KFDPARTNER` fixed view can provide information about disk partners.

```
SQL> SELECT disk "Disk", count(number_kfdpartner) "# of Partners"
FROM x$kfdpartner WHERE grp = 3 GROUP BY disk ORDER BY 1;
      Disk       # of Partners
---------- ------------------
         0                  4
         1                  4
         2                  4
         3                  4
         4                  4
         5                  4
```

ORACLE

Oracle ASM implements data redundancy by distributing mirrored copies of a disk's extents on partner disks that are located in different failure groups. Each ASM disk in a normal or high redundancy disk group can have up to eight partner disks, depending on the number of disks in the disk group. There are no disk partners or failure groups in an external redundancy disk group.

Consider a normal redundancy disk group with three disks and no specified failure groups. Each disk would have two partners. Disk 0 would partner with disk 1 and 2, disk 1 would partner with disk 0 and 2, and disk 2 would partner with disks 0 and 1. If an extent is written to disk 0, a copy of the extent is placed on either disk 1 or disk 2, but not both because only two copies are maintained in a normal redundancy disk group.

In the case of a high redundancy disk group with the same number of disks, the same partner disk relationship would exist but each extent written to a disk would result in a copy placed on each of the other two disks because three copies are maintained in a high redundancy disk group.

Now take a look at the number of disks in the `DATA2` disk group:

```
SQL> select count(disk_number) FROM v$asm_disk WHERE group_number=3;
COUNT(DISK_NUMBER)
------------------
                 6
```

Use X$KFDPARTNER to determine the number of partners that each disk in the disk group has:

```
SQL> SELECT disk "Disk", count(number_kfdpartner) "Number of partners" FROM
x$kfdpartner WHERE grp=3 GROUP BY disk ORDER BY 1;

Disk Number of partners
---------- ------------------
         0                  4
         1                  4
         2                  4
         3                  4
         4                  4
         5                  4
```

Query X$KFDPARTNER and V$ASM_DISK to determine the partner disk for each disk:

```
SQL> SELECT d.disk_number Disk#, d.failgroup Fail_Grp, p.number_kfdpartner
Partner_disk# FROM x$kfdpartner p, v$asm_disk d WHERE p.disk=d.disk_number and
p.grp=d.group_number and p.grp=3 ORDER BY 1;

     DISK# FAIL_GRP PARTNER_DISK#

---------- -------- -------------

         0 FGRP1                2
                                5
                                3

                                4
         1 FGRP1                2

                                4

                                3

                                5
         2 FGRP2                1
                                4
                                0

                                5
         3 FGRP2                4

                                0

                                5

                                1
         4 FGRP3                1

                                0

                                3
                                2
         5 FGRP3                3
                                0

                                1

                                2
```

An extent placed on any of the six disks comprising DATA2 would result in a single copy placed on one of the four partner disks.

ASM Recovery from Read I/O Errors

- When a read error by the database or ASM triggers the ASM instance to attempt bad block remapping:
  - Oracle ASM reads a good copy of the extent and copies it to the disk that had the read error
- If the write to the same location succeeds, the underlying allocation unit is deemed healthy.
- If the write fails, ASM attempts to write the extent to a new allocation unit on the same disk.
  - If this write succeeds, the original allocation unit is marked as unusable.
  - If the write fails, the disk is taken offline.

Read errors can be the result of a loss of access to the entire disk or media corruptions on an otherwise healthy disk. Oracle ASM tries to recover from read errors on corrupted sectors on a disk. When a read error by the database or Oracle ASM triggers the Oracle ASM instance to attempt bad block remapping, Oracle ASM reads a good copy of the extent and copies it to the disk that had the read error.

- If the write to the same location succeeds, the underlying allocation unit (sector) is deemed healthy. This might be because the underlying disk did its own bad block reallocation.
- If the write fails, Oracle ASM attempts to write the extent to a new allocation unit on the same disk. If this write succeeds, the original allocation unit is marked as unusable. If the write fails, the disk is taken offline.

One unique benefit of Oracle ASM–based mirroring is that the database instance is aware of the mirroring. For many types of logical corruptions such as a bad checksum or incorrect System Change Number (SCN), the database instance proceeds through the mirror side looking for valid content and proceeds without errors. If the process in the database that encountered the read can obtain the appropriate locks to ensure data consistency, it writes the correct data to all mirror sides.

ASM Recovery from Write I/O Errors

When the ASM instance receives a write error message:

1. ASM consults the Partner Status Table to see whether any of the disk's partners are offline
2. If too many partners are offline, ASM forces the dismounting of the disk group
3. Otherwise, ASM takes the disk offline

ORACLE

When the Oracle ASM instance receives a write error message from a database instance or when an Oracle ASM instance encounters a write error itself, the ASM instance attempts to take the disk offline. Oracle ASM consults the Partner Status Table (PST) to see whether any of the disk's partners are offline. If too many partners are offline, Oracle ASM forces the dismounting of the disk group. Otherwise, it takes the disk offline.

The ASMCMD `remap` command was introduced to address situations where a range of bad sectors exists on a disk and must be corrected before Oracle ASM or database I/O.

Proactively Validating Data Integrity

In earlier versions, data was checked for logical consistency when it was read.

- If a logical corruption was detected:
  – Automatic recovery was performed by using the mirror copies
  – Manual recovery was also possible by using RMAN
- For seldom-accessed data, corrupted data could be present in the system for a long time between reads.
  – Possibility that all mirrors are corrupted increases over time.

With Oracle Database 12*c*, data can be proactively scrubbed.

- Areas can be scrubbed on demand.
- Scrubbing occurs automatically during rebalance operations.

In earlier Oracle Database versions, when data was read, checks were performed on data to validate its logical consistency. If a logical corruption was detected, ASM could automatically recover by reading the mirror copies on `normal` and `high` redundancy disk groups.

One problem with this approach is that corruption of seldom-accessed data could go unnoticed in the system for a long time between reads. Also, the possibility of multiple corruptions affecting all mirror copies of data increases over time; so seldom-accessed data may simply be unavailable when it is required.

Oracle Database 12*c* introduces proactive scrubbing capabilities that check for logical corruptions and automatically repair them, where possible. In release 12.1, scrubbing can occur in two different ways:

- Scrubbing can occur as part of a rebalance operation.
- On-demand scrubbing can be performed on specific areas by an administrator.

Proactive Content Checking During Rebalance

Data that is read during rebalance is scrubbed.

- If enabled, checks are automatic with automatic error correction for mirrored data.
- Checks are enabled with the disk group attribute `content.check`.
  - Configuration example:

```
SQL> ALTER DISKGROUP DATA
     SET ATTRIBUTE 'content.check' = 'TRUE';
```


Earlier, when an ASM extent was moved during a rebalance operation, it was read and written without any additional content checks. With Oracle Database 12*c* ASM, extents that are read during rebalance can be scrubbed to ensure their logical integrity.

Rebalance scrubbing can be enabled or disabled for a disk group by using the `content.check` disk group attribute. An example of this attribute's setting is shown in the slide. By default, rebalance scrubbing is disabled (`content.check=FALSE`).

On-Demand Scrubbing

- Use the `ALTER DISKGROUP ... SCRUB` command to perform on-demand scrubbing.
  - Examples:

```
SQL> ALTER DISKGROUP DATA SCRUB REPAIR;
```

```
SQL> ALTER DISKGROUP DATA SCRUB FILE
  2  '+DATA/ORCL/DATAFILE/SYSTEM.270.775354873'
  3  REPAIR WAIT;
```

```
SQL> ALTER DISKGROUP DATA SCRUB DISK DATA_0000
  2  REPAIR POWER MAX FORCE;
```

- On-demand scrubbing operations can be monitored by using the `V$ASM_OPERATION` view.

On-demand scrubbing can be performed by using the `ALTER DISKGROUP ... SCRUB` command. On-demand scrubbing can be performed on a disk group or on individual files or individual disks. An example of each command is shown in the slide. The progress of on-demand scrubbing operations can be monitored by using the `V$ASM_OPERATION` view.

Following are details on the various options associated with on-demand scrubbing:

- If the `REPAIR` option is not specified, the specified disk group, file, or disk is only checked and any logical corruptions are reported.
- The `POWER` option can be manually set to `LOW`, `HIGH`, or `MAX`. If the `POWER` option is not specified, the scrubbing power is automatically controlled based on the system I/O load.
- If the `WAIT` option is not specified, the operation is added into the scrubbing queue and the command returns immediately. If the `WAIT` option is specified, the command returns after the scrubbing operation is completed.
- If the `FORCE` option is specified, the command is processed immediately regardless of the system I/O load.

Errors During Scrubbing

- Error messages associated with ASM scrubbing:

```
ORA-15xxx:  Logical corruption detected at [file, extent number, block#][disk, au]
```

  – The preceding error is accompanied by the following one if the `REPAIR` option is specified, but the repair is not successful.

```
ORA-15xxx:  Logical corruption cannot be repaired
```

  – More details on the corruption and the reason for the failed repair attempt are written to the trace file.

```
ORA-15xxx:  Logical corruption checking request was denied
```

  – An on-demand scrubbing request is denied because the I/O load of the system is high, or scrubbing is disabled.

Errors encountered during the scrubbing process are logged to trace files. The slide lists the new error messages that are associated with scrubbing.

Checking the Consistency of Disk Group Metadata

You can check the internal consistency of disk group metadata by using the ALTER DISKGROUP statement.

```
ALTER DISKGROUP diskgroup_name CHECK [ REPAIR | NOREPAIR ];
```

- This is conceptually the same as fsck in Linux or UNIX.
- An error summary is returned by the statement.
- Error details are written to the ASM instance alert log.
- Use the REPAIR option to resolve inconsistencies.
  - Similar to fsck -y in Linux or UNIX
- Additional specific CHECK clauses have been deprecated.

You can check the internal consistency of disk group metadata by using the ALTER DISKGROUP statement with the CHECK keyword. The disk group must be mounted to perform these checks. The statement displays summary errors and details of the errors are written to the alert log. The CHECK keyword performs the following operations:

- Verifies the consistency of the disk
- Cross-checks all the file extent maps and allocation tables for consistency
- Checks that the alias metadata directory and file directory are linked correctly
- Verifies that the alias directory tree is linked correctly
- Checks that ASM metadata directories do not have unreachable allocated blocks

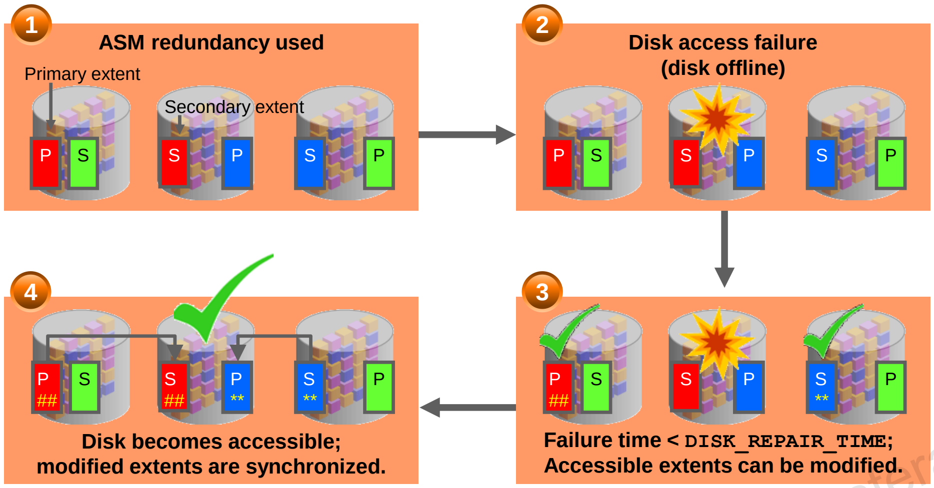
Note that these checks do not validate any stored database content such as index structures and table data.

The REPAIR clause specifies that ASM should attempt to repair errors that are found during the check. Use the NOREPAIR clause to receive alerts about inconsistencies without ASM resolving the errors automatically. The default is NOREPAIR.

In earlier releases, you could specify CHECK for ALL, DISK, DISKS IN FAILGROUP, or FILE. These additional clauses have been deprecated. Oracle recommends that you do not introduce these clauses into your new code because they are scheduled for desupport.

ASM Fast Mirror Resync

- Enabled when `COMPATIBLE.RDBMS` >= `11.1`



Whenever ASM is unable to write an extent, ASM takes the associated disk offline. If the corresponding disk group uses ASM mirroring (`normal` or `high` redundancy), at least one mirror copy of the same extent exists on another disk in the disk group.

ASM fast mirror resync is used to efficiently deal with transient disk failures. When a disk goes offline following a transient failure, ASM tracks the extents that are modified during the outage. When the transient failure is repaired, ASM can quickly resynchronize only the ASM disk extents that were modified during the outage. Note that the tracking mechanism uses one bit for each modified extent and is very efficient.

Using ASM fast mirror resync, the failed disk is taken offline but not dropped if you have set the `DISK_REPAIR_TIME` attribute for the corresponding disk group. The setting for this attribute determines the duration of disk outages that ASM will tolerate while still being able to resynchronize after the failed disk is repaired. The default setting for the `DISK_REPAIR_TIME` attribute is 3.6 hours. If a disk remains offline longer than the time specified by the `DISK_REPAIR_TIME` attribute, the disk is dropped from the disk group and the disk group is rebalanced.

Controlling the Resources Used by Resync

Power limit can be set for disk resync operations:

- It is conceptually similar to the power limit setting for disk group rebalance.
- The range is `1` (least resources) to `1024` (most resources).
- If it is not specified, the default setting is `1`.
- Examples:

```
SQL> ALTER DISKGROUP DATA ONLINE DISK data_0000 POWER 100;
```

```
ASMCMD> online -G DATA -D data_0000 --power 100
```

```
SQL> ALTER DISKGROUP DATA ONLINE ALL POWER 500;
```

```
ASMCMD> online -G DATA -a --power 500
```


ASM allows administrators to control the amount of resources that are dedicated to disk resync operations. This is conceptually similar to the capability in earlier versions, which allowed administrators to control the amount of resources dedicated to a disk group rebalance operation.

To control the amount of resources dedicated to disk resync operations, administrators can specify a power limit setting that is an integer value between 1 and 1024. Because lower numbers dedicate few resources, the operation takes longer but has minimal impact on other work. Higher values allow the operation to finish quicker at the cost of potentially impacting other work.

To specify the power limit setting for a resync operation, use the `ALTER DISKGROUP` SQL command or the ASMCMD `ONLINE` command. Examples of both commands are shown in the slide.

Resync Checkpoint and Auto-Restart

- Resync operations can be interrupted. For example:
  – A disk group is dismounted by using the `FORCE` option.
  – An ASM instance fails.
- In earlier versions:
  – Administrators had to manually re-execute the affected command
  – The entire resync operation had to be re-executed
- With Oracle Database 12*c* ASM:
  – Interrupted resync operations are automatically restarted
  – Resync operations are broken into phases
    – A checkpoint marks the end of each resync phase, and stale extent metadata is cleared.
    – Interrupted resync operations restart from the last checkpoint before the interruption.

ORACLE

A disk online operation, and an associated disk resync, could be interrupted in various ways. For example, an interruption could occur because a disk group was dismounted with the `force` option or an entire ASM instance failed.

Previously, administrators had to manually re-issue the interrupted command. Also, the metadata that was used to identify the extents that required resyncing (the stale extents) was cleared only at the end of the resync operation. If the resync operation was interrupted for any reason, the entire operation had to be re-executed.

With Oracle Database 12*c* ASM, interrupted resync operations are automatically restarted. Also, resync operations are internally broken into numerous phases, and the stale extent metadata is cleared at the end of each phase. Now, if a resync operation is interrupted and restarted, the completed phases can be skipped and processing can recommence at the beginning of the first remaining incomplete phase.

Resync Time Estimate

- Each resync operation shown in `V$ASM_OPERATION` includes a time estimate. For example:

```
SQL> SELECT PASS, STATE, EST_MINUTES FROM V$ASM_OPERATION;

PASS      STAT EST_MINUTES
--------- ---- -----------
RESYNC    RUN            1
REBALANCE WAIT           1
COMPACT   WAIT           1
```

- The `asmcmd lsop` command also displays information from `V$ASM_OPERATION`.

```
$ asmcmd rebal --power 4 DATA2
$ asmcmd lsop
Group_Name  Pass       State  Power  EST_WORK  EST_RATE  EST_TIME
DATA2       RESYNC     DONE   4      0         0         0
DATA2       COMPACT    REAP   4      0         0         0
DATA2       REBALANCE  DONE   4      6         0         0
```


There are separate rows in the `V$ASM_OPERATION` table for different phases of rebalance: disk resync, rebalance, and data compaction. The slide shows an example of the new time estimates that accompany the resync operations that are displayed in the `V$ASM_OPERATION` view.

Dealing with Transient Failure on a Failure Group

Administrators now have the option to specify a failure group repair time:

- Similar to existing disk repair time
- New disk group attribute, `failgroup_repair_time`
  - Default setting is 24 hours.
  - Configuration examples:

```
SQL> ALTER DISKGROUP DATA
     SET ATTRIBUTE 'failgroup_repair_time' = '48h';
```

```
SQL> CREATE DISKGROUP DATA2 HIGH REDUNDANCY
     FAILGROUP FG1 DISK '/dev/disk1*'
     FAILGROUP FG2 DISK '/dev/disk2*'
     FAILGROUP FG3 DISK '/dev/disk3*'
     ATTRIBUTE 'failgroup_repair_time' = '12h';
```

```
$ asmcmd setattr -G DATA failgroup_repair_time 36h
```

ORACLE

When individual disks fail, the failure is often terminal and the disk must be replaced. When all the disks in a failure group fail simultaneously, it is unlikely that all the disks individually failed at the same time. Rather it is more likely that some transient issue caused the failure. For example, a failure group could fail because of a storage network outage, a network attached storage (NAS) device rebooting, or a software crash on a storage server.

Failure group outages are more likely to be transient in nature, and replacing all the disks in a failure group is a much more expensive operation than replacing a single disk. Therefore, it would typically make sense for failure groups to have a larger repair time to ensure that all disks are not dropped automatically in the event of a failure group outage.

With Oracle Database 12*c*, ASM provides administrators with the ability to specify a failure group repair time in addition to the disk repair time that was previously available. The failure group repair time can be set by using a new disk group attribute, `FAILGROUP_REPAIR_TIME`. The default failure group repair time is 24 hours.

Now by default, if all the disks in a failure group fail simultaneously, ASM will not drop the disks when the disk repair time expires after 3.6 hours. Rather, the administrator will have 24 hours to correct the failure. If the failure is corrected before the time expires, the disks are updated by using a fast mirror resync. If the failure is not corrected in time, ASM must automatically drop all the disks in the failure group.

Even Read

In earlier versions:

- By default, ASM always reads the primary copy of the mirrored data if it is available.

With Oracle Database 12*c* ASM:

- Even Read distributes data reads evenly across all disks.
  – Each read request is sent to the least-loaded available disk.
  – Even Read is transparent to applications and enabled by default in non-Exadata environments.
  – Users on I/O-bound systems should notice a performance improvement.
- Preferred read failure groups can still be configured
  – Using the `PREFERRED_READ.ENABLED` disk group attribute
  – Using the `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter for backward compatibility

ORACLE

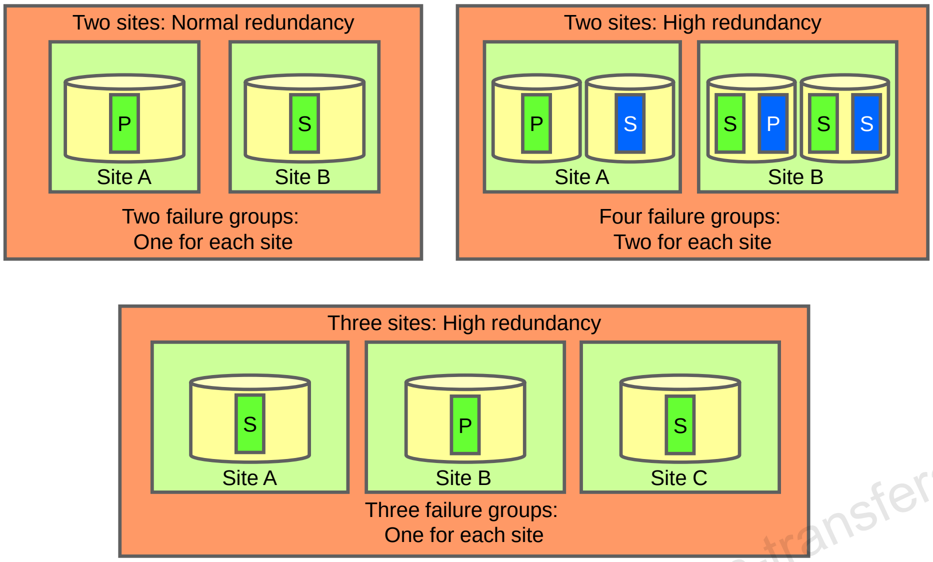
In earlier versions, the default behavior for ASM is to always read the primary copy of a mirrored extent unless a failure condition requires otherwise.

The Even Read feature of ASM, introduced in release 12.1, distributes data reads evenly across all the disks in a disk group. For each I/O request presented to the system, one or more disks may contain the data. With Even Read enabled, each read request is sent to the least loaded of the available disks.

Even Read is enabled by default on all release 12.1 (and later) database and ASM instances in non-Exadata environments. Because Even Read is transparent to applications, users on I/O-bound systems should notice a performance improvement after upgrading to release 12.1.

In an Oracle extended cluster, which contains nodes that span multiple physically separated sites, the `PREFERRED_READ.ENABLED` disk group attribute controls whether preferred read functionality is enabled for a disk group. If preferred read functionality is enabled, then this functionality enables an instance to determine and read from disks at the same site as itself, which can improve performance. Whether or not `PREFERRED_READ.ENABLED` has been enabled, preferred read can be set at the failure group level on an Oracle ASM instance or a client instance in a cluster with the `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter, which is available for backward compatibility.

Preferred Read Failure Groups: Best Practice



In practice, there are only a limited number of sensible disk group configurations in an extended cluster. A good configuration takes into account both performance and availability of a disk group in an extended cluster. Here are some possible examples:

For a two-site extended cluster, a normal redundancy disk group should have only two failure groups; all disks local to each site should belong to the same failure group. Also, no more than one failure group should be specified as a preferred read failure group by each instance. If there are more than two failure groups, ASM may not mirror extents across both sites and you will not be protected against possible site failure.

If the disk group is to be created as a high redundancy disk group, at most two failure groups should be created on each site. This guarantees that at least one extent copy is located at each site. Both local failure groups should be specified as preferred read failure groups for the local instance.

For a three-site extended cluster, a high redundancy disk group with three failure groups should be used. Thus, ASM can guarantee that each extent has a mirror copy that is local to each site and that the disk group is protected against a catastrophic disaster on any of the three sites.

Viewing ASM Disk Statistics

- `V$ASM_DISK_IOSTAT` displays information about disk I/O statistics for each ASM client.
- If `V$ASM_DISK_IOSTAT` is queried from a database instance, only the rows relating to that instance are shown.
- `ASMCMD lsdsk --statistics` shows only the disk statistics.
- `ASMCMD iostat` shows a subset of the statistics depending on the option.

ORACLE

`V$ASM_DISK_IOSTAT` relates information about disk I/O statistics to each database instance that is a client of ASM. The following example shows the use of `V$ASM_DISK_IOSTAT` to summarize I/O statistics for each instance and disk group combination:

```
SQL> SELECT INSTNAME, G.NAME DISKGROUP, SUM(READS) READS,
  2     SUM(BYTES_READ) BYTES_READ, SUM(WRITES) WRITES,
  3     SUM(BYTES_WRITTEN) BYTES_WRITTEN
  4   FROM V$ASM_DISK_IOSTAT I, V$ASM_DISKGROUP G
  5   WHERE I.GROUP_NUMBER = G.GROUP_NUMBER
  6   GROUP BY INSTNAME, G.NAME;
INSTNAME DISKGROUP READS   BYTES_READ  WRITES  BYTES_WRITTEN
-------- --------- ------- ----------- ------- -------------
orcl_3   FRA       730     12373504    20303   382233088
orcl_3   DATA      73619   1272131584  53549   953606656
```

The following query quickly shows the presence of any I/O errors reported by ASM. If this query returns a value other than zero, further investigation may be undertaken:

```
SQL> SELECT SUM(READ_ERRS)+SUM(WRITE_ERRS) ERRORS FROM V$ASM_DISK;

    ERRORS
----------
         0
```

The ASMCMD `lsdsk --statistics` command shows only the statistics columns.

```
$ asmcmd lsdsk --statistics
Reads   Write  Read_Errs  Write_Errs    Read_time     Write_Time
Bytes_Read  Bytes_Written  Voting_File  Path
 10285  250568         0          0   1175.25688  235786.607604
132737536     3275321856            Y  /dev/asmdisk1p1
 36735  257460         0          0  1535.978118  242675.490878
554306560     3337497088            Y  /dev/asmdisk1p10
 41192  453844         0          0  1359.031172  163571.150454
468881408     6068951552            Y  /dev/asmdisk1p11
  9266   54935         0          0  1100.887439  187403.175707
141057536     2347545600            N  /dev/asmdisk1p12
```

The ASMCMD `iostat` command has more ways to see the statistics, including errors.

```
$ asmcmd iostat -e
Group_Name  Dsk_Name   Reads        Writes      Read_Err  Write_Err
DATA        DATA_0000  132737536    3275420160  0         0
DATA        DATA_0001  554306560    3337630720  0         0
DATA        DATA_0002  468881408    6069426688  0         0
DATA        DATA_0003  141057536    2347548160  0         0
DATA        DATA_0008  10197078528  5199068672  0         0
DATA        DATA_0009  260273152    2244514816  0         0
FRA         FRA_0000   20326400     4323255296  0         0
FRA         FRA_0001   7868928      1492328448  0         0
FRA         FRA_0002   7997440      4240050176  0         0
FRA         FRA_0003   1105920      3528660992  0         0
```

Performance, Scalability, and Manageability Considerations for Disk Groups

- Create separate disk groups for database files and the fast recovery area.
- The disks in a disk group should have the same size and performance characteristics.
  - Allows the disk group to deliver consistent performance
  - Allows ASM to use disk space most effectively
  - Allows operations with different storage requirements to be matched with different disk groups effectively
- Using separate disk groups for each database as opposed to having multiple databases in a disk group has various benefits and drawbacks.

ORACLE

There is no ideal number of disk groups. However, there are a few guiding principles that can help you decide how many disk groups you should create.

- Create separate disk groups for your database files and fast recovery area for backup files. This configuration allows fast recovery in case of a disk group failure.
- The disks in a disk group should have the same size and performance characteristics. If you have several different types of disks in terms of size and performance, create disk groups that contain disks with similar characteristics.
  ASM load-balances file activity by uniformly distributing file extents across all the disks in a disk group. For this technique to be effective, it is important that the disks in a disk group have similar performance characteristics. For example, the newest and fastest disks might reside in a disk group that is reserved for the database work area, and slower drives could reside in a disk group that is reserved for the fast recovery area.
- There are benefits and drawbacks associated with housing multiple databases in the same disk group as opposed to maintaining each database in a separate disk group.
  Housing multiple databases in a single disk group affords the most efficient use of space. However, any faults or maintenance that affects the disk group may affect many databases. Separate disk groups provide greater isolation from the effects of a fault or maintenance operation. However, achieving this may consume more disk space and may require more disk group maintenance to balance disk resources.

Quiz

Which of the following statements about the ASM disk group Partner Status Table (PST) is NOT true?

a. The Partner Status Table stores information about the ASM disks contained in a disk group.
b. Allocation Unit 1 on each disk is reserved for the PST.
c. All the disks in a disk group will store a copy of the PST.
d. The PST must be available before a disk group can be mounted.

Answer: c

Quiz

ASM fast mirror resync is enabled when `COMPATIBLE.RDBMS` is set to 11.1 or later.

a. True

b. False

Answer: a

Fast mirror resync is enabled when `COMPATIBLE.RDBMS >= 11.1`.

Summary

In this lesson, you should have learned how to:

- Explain capacity management for external, normal, and high redundancy disk groups.
- Describe the concept of partner status tables and partner disks.
- Explain the functionality of ASM Fast Mirror Resync.
- View important ASM disk statistics.
- Explain key performance and scalability considerations for ASM disk groups

Practice 5 Overview: Administering ASM Disk Groups

This practice covers the following topics:

- Disk group space management
- Disk group administration tasks

6

Flex ASM Disk Group

Objectives

After completing this lesson, you should be able to:

- Create, implement, and manage flex disk groups
- Create, implement, and manage file groups and file group quotas

ORACLE

ASM Database-Oriented Storage Management

- Oracle ASM provides database-oriented storage management with flex disk groups.
- An Oracle ASM flex disk group is a disk group type that supports Oracle ASM file groups.
- An Oracle ASM file group is a group of files belonging to a single database.
- This association allows storage management to be performed at the database or file group level.

Oracle ASM provides database-oriented storage management with flex disk groups. An Oracle ASM flex disk group is a disk group type that supports Oracle ASM file groups. An Oracle ASM file group describes a group of files that belong to a database, and enables storage management to be performed at the file group or database level.

In general, a flex disk group enables users to manage storage at the granularity of the database, in addition to at the disk group level.

ASM Flex Disk Group Characteristics

- You can specify different redundancy and rebalance settings for different file groups.
- The redundancy of a flex disk group is set to `FLEX REDUNDANCY` and each member file group has its own redundancy property setting.
- Use `CREATE DISKGROUP` to create a flex disk group.

```
SQL> CREATE DISKGROUP flexdg1 FLEX REDUNDANCY DISK /dev/asmdisk*
```

- To migrate a disk group from `NORMAL` to `FLEX`, use:

```
SQL> ALTER DISKGROUP data MOUNT RESTRICTED;
SQL> ALTER DISKGROUP data CONVERT REDUNDANCY TO FLEX;
```

- The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` disk group attributes must be set to 12.2 or later.
- The default size of the allocation unit (AU) is 4 MB.

ORACLE

The redundancy of files in a flex disk group is flexible, enabling management at the database level. Each database has its own file group, and storage management can be done at the file group level, in addition to the disk group level. For example, you can specify different redundancy and rebalance settings for different file groups. File groups are associated with a quota group, enabling easy quota management.

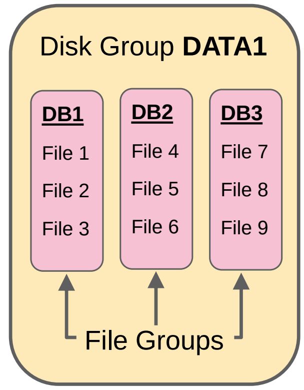
The redundancy setting of a flex disk group is set to `FLEX REDUNDANCY` and each file group of the flex disk group has its own redundancy property setting. A flex disk group generally tolerates two failures, which is the same as a `HIGH` redundancy disk group. However, if the disk group has fewer than five failure groups, it can tolerate only one failure. This restriction is the result of the quorum requirement of the disk group.

The failure tolerated by a database is dependent on the redundancy setting of its associated file group in the flex disk group. A flex disk group requires a minimum of three failure groups. Migrating to a flex disk group can be performed for a `NORMAL` or `HIGH` redundancy disk group with the `CONVERT REDUNDANCY TO FLEX` clause of the `ALTER DISKGROUP` SQL statement.

You cannot migrate an `EXTERNAL` redundancy disk group to a `FLEX` redundancy disk group. When migrating to a flex disk group, the disk group must have a minimum of three failure groups and must be mounted in restricted mode (`MOUNTED RESTRICTED`) for the duration of the migration. Virtually Allocated Metadata is enabled and required for a flex disk group. The default size of the allocation unit is 4 MB. The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` disk group attributes must be set to 12.2 or later.

DB-Oriented Storage Management with ASM File Groups

- An Oracle ASM file group is a group of files that share the same set of properties.
- This set of properties is described as the availability specification.
  - It includes mirroring, priority, quota, rebalancing, redundancy, and restoration.
- A file group is contained inside a single disk group and is dedicated to a single database, CDB, or PDB.
- A database, CDB, or PDB can have only one file group for each disk group.



An Oracle ASM file group is a group of files that share the same set of properties. This set of properties is described as the availability specification, which includes properties such as mirroring, priority, quota, rebalancing, redundancy, and restoration.

A file group is contained inside a single disk group and is dedicated to a single database, CDB, or PDB. If the database is placed across an entire disk group, then there would be one file group per disk group. A database, CDB, or PDB can have only one file group for each disk group.

The primary benefit of file groups is the ability to have a different availability specification for each database that shares the same disk group.

ASM File Group Considerations

Some important file group considerations include the following:

- A disk group contains at least one file group: the default file group.
- A disk group can contain multiple file groups.
- A disk group can store files belonging to multiple databases, with each database having a separate file group.
- A database can have only one file group in a disk group.
- A file group can belong to only one disk group.
- A file group can describe only one database, PDB, CDB, volume, or cluster.
- File groups can be created only on disk groups with the `COMPATIBLE.ASM` disk group attribute set to 12.2 or later.

The following list provides important considerations about file groups:

- A disk group contains at least one file group: the default file group.
- A disk group can contain multiple file groups.
- A disk group can store files belonging to multiple databases, with each database having a separate a file group.
- A database can have only one file group in a disk group.
- A database can span multiple disk groups with multiple file groups in different disk groups. The file groups belonging to a database in multiple disk groups should all have the same name for consistency and easy identification.
- A file group can belong to only one disk group.
- A file group can describe only one database, PDB, CDB, volume, or cluster.
- Because an Oracle ASM disk group has a limit of one million files, the number of file groups in a disk group must be less than or equal to one million.
- File groups can be created only on disk groups with the `COMPATIBLE.ASM` disk group attribute set to 12.2 or later.

Oracle ASM Quota Groups

- A quota group defines the quota allocated to a group of Oracle ASM file groups.
- A file group can belong to only one quota group.
- A quota can describe an aggregate of space used by different databases.
- A quota is enforced at the file group level, not at the user level.
- Add a quota group to a disk group.

```
ALTER DISKGROUP data2 ADD QUOTAGROUP quota_grp_B SET quota= 100G;
```

- Add a file group to a quota group.

```
ALTER DISKGROUP data2 MODIFY FILEGROUP fg_pdb1
SET quota_group = quota_grp_B;
```

ORACLE

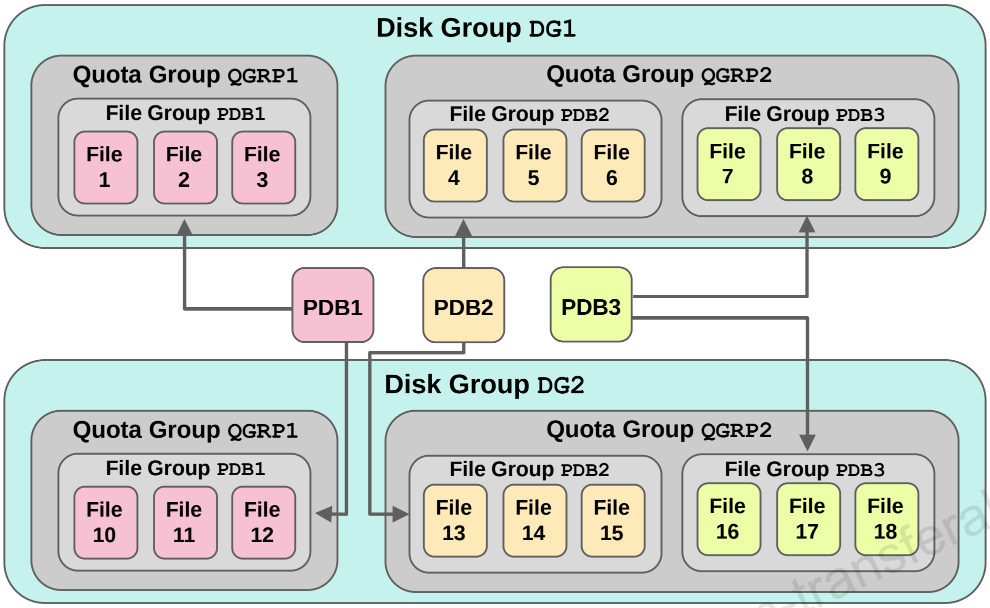
A quota group defines the quota allocated to a group of Oracle ASM file groups. A file group belongs to only one quota group. A quota can describe an aggregate of space used by different databases. A quota is enforced at the file group level instead of the user level.

The following list provides important considerations about quota groups for file groups.

- A file group can belong to only one quota group.
- A quota group cannot span multiple disk groups.
- A quota group describes an aggregate of space used by one file group or multiple file groups in the same disk group.
  A file group for a PDB and a file group for a clone of that PDB could be a scenario where file groups would share a common quota group.
  A file group clone by default inherits the parent's quota group. However, the clone could belong to a different quota group.

You can administer quota groups with SQL and ASMCMD commands. The examples in the slide illustrate how SQL statements are used to add a quota group to a disk group and how to add a file group to a quota group. You can monitor ASM quota groups by using the `V$ASM_QUOTAGROUP` dynamic view.

ASM File Groups Deployment: Example



The example in the slide illustrates the use of Oracle ASM file groups and quota groups in a multitenant environment. In this illustration, PDB3 is assumed to be a clone of PDB2. File groups and quota groups in different disk groups can have the same name.

File groups named PDB1 in disk group DG1 and disk group DG2 are dedicated to pluggable database PDB1; file groups named PDB2 in disk group 1 and disk group 2 are dedicated to pluggable database PDB2; and file groups named PDB3 in disk group DG1 and disk group DG2 are dedicated to pluggable database PDB3.

File group PDB1 in disk group DG1 belongs to quota group QGRP1 in disk group DG1. File group PDB1 in disk group DG2 belongs to quota group QGRP1 in disk group DG2. File groups PDB2 and PDB3 in disk group DG1 belong to quota group QGRP2 in disk group DG1. File groups PDB2 and PDB3 in disk group DG2 belong to quota group QGRP2 in disk group DG2.

Oracle ASM File Group Properties

The following list summarizes the file group properties of file groups:

- PRIORITY
- QUOTA_GROUP
- POWER_LIMIT
- COMPATIBLE
- OWNER
- PENDING_REMIRROR
- LAST_REMIRROR_TS

ORACLE

The following list summarizes the file group properties of file groups.

- PRIORITY: This file group property specifies the rebalance priority for restore. File groups with the highest priority are relocated first, then file groups with the next highest priority are relocated, and so on. Valid settings are HIGHEST, HIGH, MEDIUM, LOW, and LOWEST. The default value is MEDIUM. The COMPATIBLE.RDBMS disk group attribute must be set to 11.2 or later.
- QUOTA_GROUP: This file group property specifies the name of the quota group to which the file group belongs. A valid setting is an existing quota group name. The default value is GENERIC. Every file group in the disk group must be a member of a quota group. The COMPATIBLE.RDBMS disk group attribute must be set to 11.2 or later.
- POWER_LIMIT: This file group property specifies the rebalance power, which determines the number of parallel relocations for disk rebalancing of the file group. Valid settings are 0 through 1024. The default value is 1. The COMPATIBLE.RDBMS disk group attribute must be set to 11.2 or later.
- COMPATIBLE: This is a file group property. The current value is the last known client compatibility during a file open operation. For CLUSTER and GENERIC file groups, the file group COMPATIBLE value is NULL, which is the default value. The COMPATIBLE.RDBMS disk group attribute must be set to 12.1.0.2 or later. COMPATIBLE.RDBMS must be set less than or equal to the Oracle Database COMPATIBLE initialization parameter and both must be set less than or equal to COMPATIBLE.ASM.

  COMPATIBLE.RDBMS cannot be advanced beyond the minimum compatibility of all file groups for the client type DATABASE on the disk group. Use VOLUME FORCE to update the COMPATIBLE.RDBMS property so that no one can set the file group COMPATIBLE setting by using ALTER FILEGROUP.
- OWNER: This is a file group property. The setting must be a valid operating system username that can be contained in a field of VARCHAR2(64). The default is NULL. The COMPATIBLE.RDBMS disk group attribute must be set to 11.2 or later.

- PENDING_REMIRROR: This is a file group property that keeps track whether a file group needs remirroring. This is a hidden property. The COMPATIBLE.RDBMS disk group attribute must be set to 12.2 or later.
- LAST_REMIRROR_TS: This is a file group property that keeps track of the time stamp when remirroring was done last. This is a hidden property. The COMPATIBLE.RDBMS disk group attribute must be set to 12.2 or later.

ASM File Group Administration by Using SQL

- Adding a file group to a database:

```
ALTER DISKGROUP data1 ADD FILEGROUP fg_pdb1 DATABASE pdb1
SET quota_group = quota_grp_pdb1;

ALTER DISKGROUP data1 ADD FILEGROUP fg_pdb2 DATABASE pdb2
SET quota_group = quota_grp_pdb1;
```

- Modifying a file group:

```
ALTER DISKGROUP data1 MODIFY FILEGROUP fg_pdb1
SET 'controlfile.redundancy' = 'HIGH';
```

```
ALTER DISKGROUP data1 MODIFY FILEGROUP fg_pdb1
SET 'datafile.redundancy' = 'HIGH';
```

```
ALTER DISKGROUP data1 MODIFY FILEGROUP fg_pdb1
SET 'archivelog.redundancy' = 'MIRROR';
```

ORACLE

You can administer file groups with SQL and ASMCMD commands. The examples shown in the slide illustrate how SQL statements are used to add a file group to a database and how to modify a file group. In this example, pluggable database pdb2 is a clone of pdb1.

Using ASMCMD to Manage File Groups and Quota Groups

ASMCMD file group and quota group commands:

- `chfg`: Modifies a file group based on an XML configuration file or script
- `chqg`: Modifies a quota group in the disk group
- `lsfg`: Lists the file groups in a disk group
- `lsqg`: Lists the quota groups in a disk group
- `mkfg`: Creates a file group in a disk group based on an XML configuration file or script
- `mkqg`: Adds a quota group to the disk group
- `mvfg`: Moves a file group to the specified quota group
- `rmfg`: Removes an existing file group from a disk group
- `rmqg`: Drops a quota group from a disk group

ORACLE

The following is an example of an XML configuration file for `chfg`. The configuration file alters a file group named filegroup1 on disk group dg_data1. The file group is associated with quota group quotagroup1. The files in this file group have the redundancy set to mirror, except the control file, which has high redundancy. All other attributes are not changed.

```
<filegroup name="filegroup1" dg="dg_data1">
 <p name="redundancy" value="mirror"/>
 <p name="redundancy" value="high" file_type="controlfile"/>
 <p name="quota_group" value="quotagroup1"/>
</filegroup>
```

The following is an example of an XML configuration file for `mkfg`. The configuration file creates a file group named filegroup1 on disk group dg_data1 for a database named `sample`. The file group is associated with quota group quotagroup1. The files in this file group have the redundancy set to mirror, except the control file, which has high redundancy. All other attributes are set to the default value.

```
<filegroup name="filegroup1" dg="dg_data1" client_type="database"
client_name="sample">
 <p name="redundancy" value="mirror"/>
 <p name="redundancy" value="high" file_type="controlfile"/>
 <p name="quota_group" value="quotagroup1"/>
</filegroup>
```

New ASM File Group Dynamic Views

- `V$ASM_FILEGROUP`
- `V$ASM_FILEGROUP_FILE`
- `V$ASM_FILEGROUP_PROPERTY`

Several new dynamic views have been included to monitor Oracle ASM file groups. These include:

- `V$ASM_FILEGROUP`: Contains one row for every file group in every disk group mounted by the Oracle ASM instance
- `V$ASM_FILEGROUP_FILE`: Contains one row for each file associated with a file group in every disk group mounted by the ASM instance
- `V$ASM_FILEGROUP_PROPERTY`: Contains one row for every property associated with a file group or each file type of every file group in every disk group mounted by the Oracle ASM instance

Note that all `V$ASM_FILESET`* views have been deprecated in Oracle Database 12.2.

Prioritized Rebalancing for File Groups

- Oracle ASM support for ASM file groups includes the ability to set priority levels for rebalancing of file groups.
- Use the `ALTER DISKGROUP... FILEGROUP` statement with a `PRIORITY` clause.
- Set `PRIORITY` to `HIGHEST`, `HIGH`, `MEDIUM`, `LOW`, or `LOWEST` to ensure that certain types of files are relocated before others.

```
ALTER DISKGROUP data1 MODIFY FILEGROUP fg_pdb1
SET PRIORITY = HIGH;
```

- If `PRIORITY` is not explicitly specified for a file group, the value defaults to `MEDIUM`.

Oracle ASM support for Oracle ASM file groups includes the ability to set priority levels for rebalancing of file groups. The `ALTER DISKGROUP` SQL statement is updated with a `FILEGROUP` clause.

You can set the rebalance priority for restore to `HIGHEST`, `HIGH`, `MEDIUM`, `LOW`, or `LOWEST` to ensure that certain types of files are relocated before others. File groups with the highest priority are completely relocated first, then file groups with the next highest priority are relocated, and so on. `MEDIUM` is the default. The RDBMS compatibility disk group attribute must be set to 11.2 or later.

Summary

In this lesson, you should have learned how to:

- Create, implement, and manage flex disk groups
- Create, implement, and manage file groups and file group quotas

Practice 6 Overview: Flex Disk Groups

This practice covers the following topics:

- Working with Flex Disk Groups, File Groups, and Quota Groups

7

Administering ASM Files, Directories, and Templates

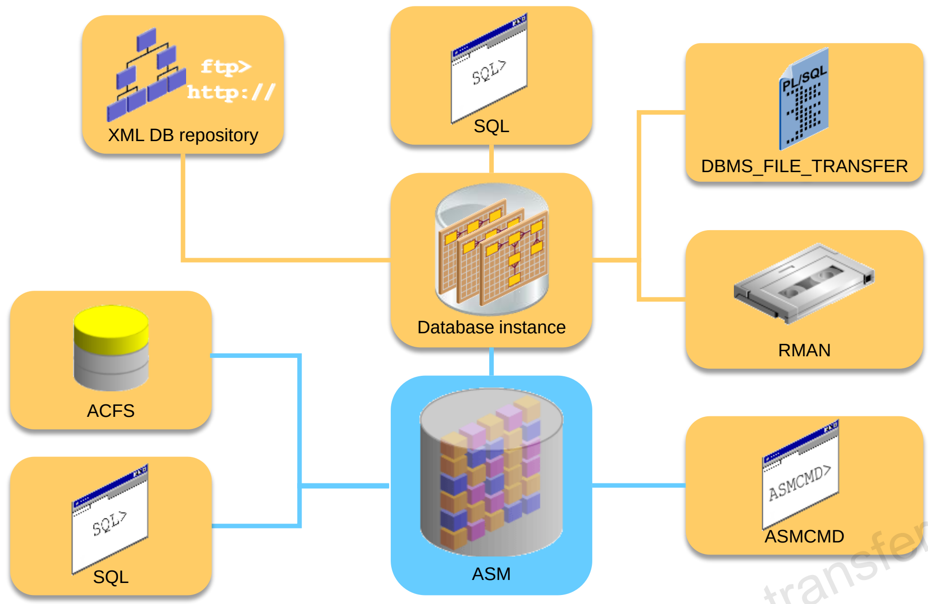
Objectives

After completing this lesson, you should be able to:

- Use different client tools to access Automatic Storage Management (ASM) files
- Describe the format of a fully qualified ASM file name
- Explain how ASM files, directories, and aliases are created and managed
- Describe and manage disk group templates

ORACLE

ASM Clients



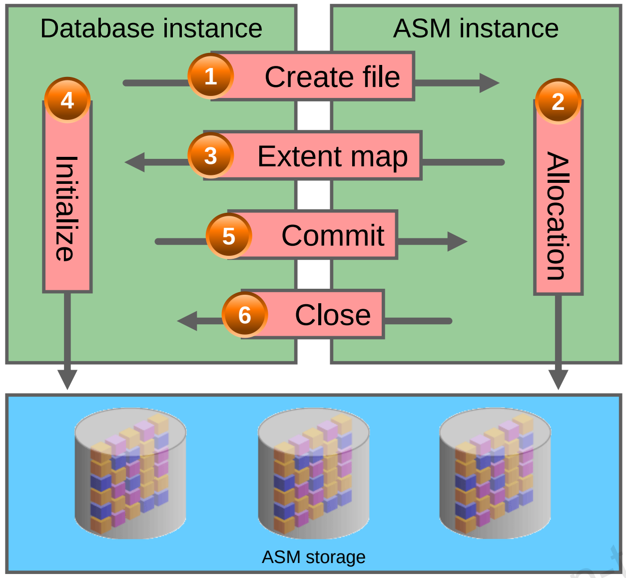
Oracle ASM is a volume manager and a file system for Oracle database files. This slide introduces various ASM clients. Many of these are examined in greater detail later in this lesson. Clients that connect directly to ASM include:

- **Oracle Database:** Oracle Database is the most fundamental ASM client. It makes requests to ASM relating to numerous types of activities such as ASM file creation.
- **ASM Cluster File System:** This depends on ASM to provide the ASM volumes.
- **ASM Clusterware:** If Oracle Cluster Registry (OCR) files, voting files, or the ASM server parameter file (SPFILE) is stored on an ASM disk group, the ASM instance is a client of itself.
- **ASMCA:** ASM Configuration Manager is a graphical interface that allows you to manage ASM instances, disk groups, and volumes.
- **Enterprise Manager:** This allows you to manage and monitor the ASM instance directly and indirectly through the database.
- **Grid Infrastructure:** This uses ASM to store the OCR and voting disk files by default.
- **SQL clients (such as SQL*Plus):** A series of ASM-specific SQL commands (such as `CREATE DISKGROUP`) provide the most fundamental management client for ASM.
- **ASMCMD:** The ASM command-line interface is used to interrogate and manage ASM. It includes many UNIX-like commands that can be used to manage the files and directories in an ASM system.

In addition to clients that connect to ASM directly, there are a series of interfaces provided by Oracle Database that can be used to manipulate ASM in different ways. These include:

- **SQL:** When an Oracle Database is managed under ASM, activities that create and delete ASM files (such as `CREATE DATABASE` and `DROP TABLESPACE`) implicitly interact with the underlying ASM instance. A database instance also directly reads and writes ASM files as a result of SQL data manipulation language (DML) commands (`INSERT`, `UPDATE`, and `DELETE`). However, operations that involve space management (such as extending a data file) also require interaction with the ASM instance.
- **Oracle Recovery Manager** (`RMAN`)**:** `RMAN` is Oracle's recommended backup and recovery utility. `RMAN` is well integrated with ASM and can be used to migrate non-ASM databases into ASM.
- **XML DB:** ASM files and directories can be accessed through a virtual folder in the XML DB repository. XML DB provides a means to access and manipulate the ASM files and directories with programmatic APIs, such as the `DBMS_XDB` package, and with XML DB protocol services such as FTP and HTTP/WebDAV.
- **`DBMS_FILE_TRANSFER`:** The `DBMS_FILE_TRANSFER` package provides procedures to copy ASM files within a database or to transfer binary files between a local ASM instance and a remote database file. `DBMS_FILE_TRANSFER.COPY_FILE` supports all transfer combinations involving ASM and/or your local file system, namely:
  - Local file system to local file system
  - Local file system to ASM
  - ASM to local file system
  - ASM to ASM

Interaction Between Database Instances and ASM



The file creation process provides an illustration of the interactions that take place between database instances and ASM. The file creation process occurs as follows:

1. The database requests file creation.
2. An ASM foreground process creates a Continuing Operation Directory (COD) entry and allocates space for the new file across the disk group.
3. A background database process (ASMB) receives an extent map for the new file.
4. The file is now open and the database process initializes the file directly.
5. After initialization, the database process requests that the file creation be committed. This causes the ASM foreground process to clear the COD entry, marking the file as created.
6. Acknowledgment of the file commit implicitly closes the file. The database instance will need to reopen the file for future I/O.

There are two important points to consider from this example. First, the database instance and ASM instance work together in a coordinated fashion. A database instance must interact with ASM to map database files to ASM extents. It also receives a stream of messages relating to ASM operations that may lock or move ASM extents. Secondly, database I/O is not channeled through the ASM instance. In fact, the database conducts I/O operations directly against ASM files, as illustrated in step 4 in the slide.

Accessing ASM Files by Using RMAN

- RMAN should be used to back up ASM files.
  - RMAN is needed to unmap data file extent locations.
  - Third-party Oracle Database backup and recovery managers also use RMAN to access ASM.
- ASM can be used as a storage area for RMAN backups.
- Depending on context, RMAN can reference individual ASM files or entire disk groups.

```
RMAN> BACKUP AS COPY
      DATAFILE "+DATA/rdbms/datafile/tbs_5.256.565313879"
      FORMAT   "+DATA2";
```

- RMAN can be used to migrate databases into ASM.
  - Migrating a database to ASM can be done one file at a time, or all at once.

ORACLE

Oracle Recovery Manager (RMAN) is the preferred method for backup and recovery of databases contained in ASM. RMAN is very well integrated with ASM and is also used by third-party Oracle Database backup and recovery managers.

RMAN can back up from and recover to ASM. It can also use ASM as a store for backup sets.

Depending on context, RMAN can reference individual ASM files or entire disk groups. For example, the following RMAN BACKUP command refers to an individual ASM file on the second line and also refers to an ASM disk group on the third line.

```
RMAN> BACKUP AS COPY
  DATAFILE "+DATA/rdbms/datafile/tbs_5.256.565313879"
  FORMAT   "+DATA2";
```

RMAN can also be used to migrate existing databases into ASM. This can be done one file at a time or a complete database can be migrated into ASM in the same operation.

The following provides an example of the procedure that you can use to relocate your entire database to an ASM disk group (assuming the use of a server parameter file):

1. Obtain the file names of the current control files and online redo logs by using V$CONTROLFILE and V$LOGFILE.
2. Shut down the database consistently.
3. Modify the server parameter file of your database as follows:
   - Start the database with the NOMOUNT option.
   - Set the DB_CREATE_FILE_DEST parameter to the desired ASM disk group.
   - Remove the CONTROL_FILES parameter. It is re-created automatically.

4. Edit to replace the placeholder file and disk group references with your actual locations, and then run the following RMAN command file. This backs up the database, switches the current data files to the backups, renames the online redo logs, and re-creates the temporary tablespaces.

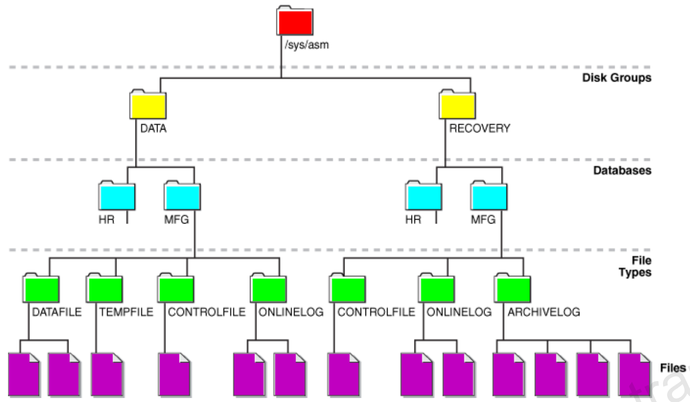
```
RESTORE CONTROLFILE FROM '/u01/c1.ctl';
ALTER DATABASE MOUNT;
BACKUP AS COPY DATABASE FORMAT '+YOURDG';
SWITCH DATABASE TO COPY;
# Repeat command for all online redo log members
SQL "ALTER DATABASE RENAME '/u01/log1' TO '+YOURDG' ";
ALTER DATABASE OPEN RESETLOGS;
# Repeat commands for all temporary tablespaces
SQL "ALTER TABLESPACE temp ADD TEMPFILE";
SQL "ALTER DATABASE TEMPFILE '/u01/temp1' DROP";
```

5. Delete the old database files.

Note: This example illustrates the procedure for migrating a database into ASM. You may want to use other options and settings to migrate your specific databases into ASM. For a complete discussion of this topic, refer to the *Oracle Database Backup and Recovery User's Guide 12c Release 1*.

Accessing ASM Files By Using XML DB

- ASM files and directories can be accessed through the `/sys/asm` virtual folder in the XML DB repository.
  - Access is via PL/SQL APIs, FTP, and HTTP/WebDAV.
- The diagram shows the hierarchy under `/sys/asm`.



ASM files and directories can be accessed through a virtual folder in the Oracle Database XML DB repository. The repository path to the virtual folder is `/sys/asm`. The folder is virtual because its contents do not actually reside in the repository; they exist as normal ASM files and directories. `/sys/asm` provides a means to access and manipulate the ASM files and directories with programmatic APIs, such as the `DBMS_XDB` package, and with XML DB protocols such as FTP and HTTP/WebDAV. You must log in to XML DB as a user other than `SYS` and that user must be granted the `DBA` role to access `/sys/asm` with XML DB protocols.

A typical use for this capability might be to view `/sys/asm` as a web folder in Windows Explorer, and then copy a Data Pump dumpset from an ASM disk group to an operating system file system by dragging and dropping.

Under `/sys/asm`, the folder hierarchy is defined by the structure of an ASM fully qualified file name. That is, the `/sys/asm` virtual folder contains one subfolder for every mounted disk group, and each disk group folder contains one subfolder for each database that uses the disk group. In addition, a disk group folder might contain files and folders corresponding to aliases created by the administrator. Continuing the hierarchy, the database folders contain ASM file type folders, which contain the ASM files.

Note: XML DB must be configured to enable FTP and HTTP/WebDAV. Use EM or the script provided at `$ORACLE_HOME/rdbms/admin/catxdbdbca.sql` to do this.

Accessing ASM Files By Using DBMS_FILE_TRANSFER

- DBMS_FILE_TRANSFER provides procedures to:
  - Copy ASM files within a database
  - Transfer binary files in either direction between a local ASM instance and a remote database file
- DBMS_FILE_TRANSFER.COPY_FILE supports all transfer combinations between ASM and the local files.
- Example:
  - Copy a local database file into ASM.

```
SQL> CREATE DIRECTORY dgb AS '+DATA/dbfiles';
SQL> CREATE DIRECTORY loc AS '/u01/app/oracle/oradata/db';
SQL> BEGIN
  2   DBMS_FILE_TRANSFER.COPY_FILE('loc','tmp.dbf','dgb','tmp.dbf');
  3 END;
  4 /
```

ORACLE

The DBMS_FILE_TRANSFER package provides procedures to copy ASM files within a database or transfer binary files between the databases that use ASM. The DBMS_FILE_TRANSFER package has the following procedures:

- COPY_FILE: Reads a file from a source directory and creates a copy of the file in a destination directory. The source and destination directories can both be in a local file system or in an ASM disk group. You can also use this procedure to copy between a local file system and an ASM disk group; the copy operation is valid in either direction.
- GET_FILE: Contacts a remote database to read a remote file, and then creates a copy of the file in the local file system or ASM disk group
- PUT_FILE: Reads a local file or ASM disk group and contacts a remote database to create a copy of the file in the remote file system

When using DBMS_FILE_TRANSFER, note the following:

- The size of the copied file must be a multiple of 512 bytes.
- The size of the copied file must be less than or equal to two terabytes.
- Transferring a file is not transactional. To guarantee consistency, bring files offline when the database is in use.
- The copied file is treated as a binary file, and no character set conversion is performed.

Accessing ASM Files By Using ASMCMD

- ASMCMD provides a command-line interface to interrogate and manage ASM.
- It includes many UNIX-like commands that can be used to manage the files and directories in an ASM system.
- Example AMSCMD session:

```
[grid@host01 ~]$ asmcmd
ASMCMD [+] > cd +DATA/orcl/datafile
ASMCMD [+DATA/orcl/datafile] > ls -l SYS*
Type      Redund  Striped  Time             Sys  Name
DATAFILE  MIRROR  COARSE   AUG 03 16:00:00  Y    SYSAUX.257.692926339
DATAFILE  MIRROR  COARSE   AUG 03 16:00:00  Y    SYSTEM.256.692926339
ASMCMD [+DATA/orcl/datafile] > cd ..
ASMCMD [+DATA/orcl] > pwd
+DATA/orcl
ASMCMD [+DATA/orcl] > exit
[grid@host01 ~]$
```

ORACLE

ASMCMD is a command-line utility that you can use to view and manipulate the files and directories within ASM disk groups. ASMCMD can list the contents of disk groups, perform searches, create and remove directories and aliases, display space utilization, and more.

When you run ASMCMD, you connect to an ASM instance and you are provided a command line where you can execute any of the `ASMCMD` commands. The `ASMCMD` commands include many commands that perform the same basic functions as their UNIX/Linux counterparts. These include: `cd`, `cp`, `du`, `find`, `ls`, `mkdir`, `pwd`, and `rm`. There are additional commands that perform ASM-specific functions.

`ASMCMD` commands are discussed in parallel with the other utilities for performing ASM administration tasks.

When you invoke the ASMCMD utility from the Grid software `$ORACLE_HOME`, the utility attempts to connect `AS SYSASM`. It attempts to connect `AS SYSASM`, so the user that invokes ASMCMD must be a member of the OSASM group.

When connecting from the `$ORACLE_HOME` database, ASMCMD attempts to connect `AS SYSDBA`. The OS user must be a member of the OSDBA group for ASM. Set the `ORACLE_SID` environment variable to the local name of the database instance. Any OS user who is a member of the appropriate OS group may connect with ASMCMD to a local instance, and specify the connect role with the `–a` option.

Fully Qualified ASM File Names

- Every file that is created in ASM gets a system-generated file name, which is known as the fully qualified file name.
  - You cannot set the fully qualified file name.
  - ASM guarantees uniqueness within the ASM environment.
  - The fully qualified file name is used in database views that display Oracle Database file names.
- Format:

```
<+group>/<db_unique_name>/<file_type>/<file_type_tag>.<file>.<incarnation>
```

- Examples:

```
+DATA/ORA12c/DATAFILE/SYSTEM.262.676172197
+DATA/ORA12c/PARAMETERFILE/spfile.268.676172697
+DATA/ORA12c/CONTROLFILE/Current.257.676172363
+DATA/ORA12c/ONLINELOG/group_1.256.676172371
+FRA/ORA12c/CONTROLFILE/Backup.275.676172942
```

ORACLE

Every file that is created in ASM gets a system-generated file name, which is known as the fully qualified file name. You cannot set the fully qualified file name. The fully qualified file name represents a complete path name in the ASM file system. An example of a fully qualified file name is:

`+DATA/orcl/DATAFILE/SYSTEM.262.676172197`

You can use the fully qualified file name to reference (read or retrieve) an ASM file. You can also use other abbreviated file name formats, such as an alias ASM file name (discussed later in this lesson) to reference an ASM file. You can find the fully qualified file name in database views that display Oracle Database file names, such as `V$DATAFILE` and `V$LOGFILE`.

A fully qualified file name has the following form:

`+group/db_unique_name/file_type/file_type_tag.file.incarnation`

where:

- `+group` is the disk group name preceded by a plus sign. You can think of the plus sign as the root of the ASM file system, similar to the slash (/) on UNIX or Linux file systems.
- `db_unique_name` is the name of the database to which the file belongs.
- `file_type` is the Oracle Database file type.
- `file_type_tag` is type-specific information about the file.
- `file.incarnation` is the file/incarnation pair, used to ensure uniqueness.

The following table shows possible values for the file_type and file_type_tag components of a fully qualified file name:

ASM File Type	ASM File Type Tag	Description
CONTROLFILE	Current or Backup	Control files and backup control files
DATAFILE	*<tsname>*	Data file and data file copies. <tsname> is the tablespace containing the data file.
ONLINELOG	group_*<group#>*	Online logs
ARCHIVELOG	thread_*<thread#>*_seq_*<sequence#>*	Archive logs
TEMPFILE	*<tsname>*	Temporary tablespace file. <tsname> is the tablespace containing the data file.
BACKUPSET	*<sp?>_<timestamp>*	Data file backup, data file incremental backup, or archive log backup pieces. <sp> has the value s when the backup set includes the spfile; n indicates that the backup set does not include the spfile.
PASSWORDFILE	*<pwfile>*	Shared password file for either the database or ASM
PARAMETERFILE	Spfile	Server parameter files
DATAGUARDCONFIG	*<db-unique-name>*	Data Guard configuration file. <dbname> is the value of the DB_UNIQUE_NAME database initialization parameter.
FLASHBACK	log_*<log#>*	Flashback logs
CHANGETRACKING	Ctf	Block change tracking file, used in incremental backups
DUMPSET	*<user>_<job#>_<file#>*	Data Pump dumpset. Dumpset files encode the username, the job number that created the dumpset, and the file number as part of the tag.
XTRANSPORT	*<tsname>*	Cross-platform transportable tablespace data files. <tsname> is the tablespace containing the data file.
AUTOBACKUP	*<sp?>_<timestamp>*	Automatically generated control file backup. <sp> has the value s when the backup set includes the spfile; n indicates that the backup set does not include the spfile.
ASMPARAMETERFILE	Registry	Name of the Oracle ASM SPFILE
OCRFILE	*<ocrfile>*	Name of the OCR files

Other ASM File Names

- Alias ASM file name

Format: `<+group>/<alias>`
Example: `+DATA/my_dir/my_other_dir/my_file_name.dbf`

- Alias ASM file name with template

Format: `<+group>(<template>)/<alias>`
Example: `+DATA(my_template)/my_dir/my_other_file_name.dbf`

- Incomplete ASM file name

Format: `<+group>`

- Incomplete ASM file name with template

Format: `<+group>(<template>)`
Example: `+DATA(my_template)`

In addition to the fully qualified file name, the following other ASM file names can be used in various different situations:

Alias ASM File Name

Alias ASM file names, or simply aliases, can be used both for referencing existing ASM files and for creating new ASM files. Alias names start with the disk group name preceded by a plus sign, after which you specify a name string of your choice. Alias file names are implemented by using a hierarchical directory structure, with the slash (/) or backslash (\) character separating the name components. You can create an alias in any system-generated or user-created ASM directory that is already in existence, except at the root (+) level.

Alias ASM file names are distinguished from fully qualified file names or numeric file names because they do not end in a dotted pair of numbers. It is an error to attempt to create an alias that ends in a dotted pair of numbers. You can think of aliases as achieving the same function as symbolic links in a UNIX or Linux environment.

Alias ASM File Name with Template

An alias ASM file name with template is essentially the same as an alias ASM file name. The only difference is that a file that is created by using this type of file name receives the mirroring and striping attributes specified by the named template. The template must belong to the disk group that the file is being created in. An alias ASM file name with template is used only for ASM file creation operations.

Incomplete ASM File Name

Incomplete ASM file names are used only for file creation operations. An incomplete ASM file name is only a reference to a disk group name. When incomplete ASM file names are used, ASM automatically creates fully qualified file names under the specified disk group.

Incomplete ASM File Name with Template

Incomplete ASM file names with templates are also used only for file creation operations. They consist of the disk group name followed by the template name in parentheses. This explicit template reference causes ASM to use the specified template to determine mirroring and striping attributes for the file, instead of the default template for that file type.

Incomplete ASM file names, with and without templates, can be used in conjunction with the Oracle Managed Files (OMF) feature of Oracle Database. If the `DB_CREATE_FILE_DEST` initialization parameter for the database is set to an incomplete ASM file name, it is possible to create a tablespace without any further reference to a file name.

For example, assume the following initialization parameter setting:

```
DB_CREATE_FILE_DEST = '+DATA'
```

The following statement creates the `tspace1` tablespace.

```
CREATE TABLESPACE tspace1;
```

ASM automatically creates and manages a data file for `tspace1` on the ASM disks in the `DATA` disk group. The file extents are stored by using the attributes defined by the default template for a data file.

Valid Contexts for the ASM File Name Forms

File Name Form	Valid to Reference Existing Files	Valid for Single File Creation	Valid for Multiple File Creation	Created as OMF
Fully Qualified	Yes	No	No	No
Alias	Yes if defined	Yes	No	No
Alias with Template	No	Yes	No	No
Incomplete	No	Yes	Yes	Yes
Incomplete with Template	No	Yes	Yes	Yes

ORACLE

The table in the slide specifies valid contexts for each file name form.

Single file creation is an operation that creates a single file, such as a data file or a control file. You can use an alias or incomplete file name, where a file name is called for in a statement to create a single file, such as `CREATE TABLESPACE` or `CREATE CONTROLFILE`.

Multiple file creation is an operation that can create multiple ASM files. An example of such an operation would be a single `ALTER DATABASE` statement that adds a new online redo log group with multiple members.

Single File Creation: Examples

- Alias ASM file name

```
SQL> ALTER TABLESPACE myspace ADD
   2 DATAFILE '+DATA/mydir/myspace02.dbf' SIZE 50M;
```

- Alias file name with template

```
SQL> ALTER TABLESPACE myspace ADD
   2 DATAFILE '+DATA(mytemplate)/mydir/myspace03.dbf';
```

- Incomplete file name

```
SQL> ALTER TABLESPACE myspace ADD
   2 DATAFILE '+DATA' SIZE 50M;
```

ORACLE

In a single file creation operation, you specify one of the following ASM file name forms instead of a local file name. As previously mentioned, the valid forms for single file creation operations are:

- Alias ASM file name
- Alias ASM file name with template
- Incomplete ASM file name
- Incomplete ASM file name with template

Note: When the `SIZE` parameter is not given, `SIZE` defaults to 100 MB and the file is auto-extensible with an unlimited maximum size.

Multiple File Creation: Example

- Create an online redo log group with two members.
  - Set the following database initialization parameters:

```
DB_CREATE_ONLINE_LOG_DEST_1 = '+DATA'
DB_CREATE_ONLINE_LOG_DEST_2 = '+FRA'
```

  - Then create the log group without any file references:

```
SQL> ALTER DATABASE ADD LOGFILE;
```

- The resulting fully qualified file names are:

```
+DATA/ORA12c/ONLINELOG/group_5.269.699798885
+FRA/ORA12c/ONLINELOG/group_5.256.699799169
```

ORACLE

Multiple file creation operations work only in conjunction with incomplete ASM file names. In such operations, the incomplete file name reference is implied through the use of the following database initialization parameters:

- `DB_CREATE_FILE_DEST`
- `DB_CREATE_ONLINE_LOG_DEST_<n>` (where <n> = 1, 2, 3, 4 or 5)

The slide example illustrates how an online redo log group containing two members is created without any explicit file references.

Although it is recommended that you use the Database Configuration Assistant (DBCA) to create databases, it is possible to set `DB_CREATE_FILE_DEST` to an ASM disk group and create a complete database by using the following simple statement:

```
CREATE DATABASE sample;
```

This statement creates a database with at least the following ASM files:

- A `SYSTEM` tablespace data file in the disk group specified in `DB_CREATE_FILE_DEST`
- A `SYSAUX` tablespace data file in the disk group specified in `DB_CREATE_FILE_DEST`
- A multiplexed online redo log with two log groups. Each log group will have one member in the disk group specified in `DB_CREATE_FILE_DEST` and a second member in the disk group specified in `DB_RECOVERY_FILE_DEST`.

Viewing ASM Aliases, Files, and Directories

- In an ASM instance, V$ASM_ALIAS displays one row for each alias, system-generated file name, and directory that is present in every disk group mounted by the ASM instance.
- V$ASM_ALIAS contains the full hierarchy of aliases, files, and directories.
  - Use a hierarchical query to show this.
- In a database instance, V$ASM_ALIAS displays aliases for the files in the database.
- The asmcmd ls command behaves much like the UNIX ls command.

V$ASM_ALIAS contains a record for each system-generated file name, user-defined alias, and directory for every currently mounted disk group.

Use a hierarchical query on V$ASM_ALIAS to reconstruct the hierarchy of files, aliases, and directories in ASM. Join V$ASM_ALIAS with V$ASM_FILE to display additional file information, such as the file type. V$ASM_ALIAS is also commonly joined with V$ASM_DISKGROUP to display the disk group name.

The asmcmd ls -l command lists the contents of a directory:

```
ASMCMD> ls -l +DATA/ORCL/DATAFILE
Type      Redund  Striped  Time             Sys  Name
DATAFILE  MIRROR  COARSE   JUL 26 22:00:00  Y    EXAMPLE.277.821807905
DATAFILE  MIRROR  COARSE   AUG 02 11:00:00  Y    SYSAUX.269.821807195
DATAFILE  MIRROR  COARSE   JUL 26 22:00:00  Y    SYSTEM.270.821807311
DATAFILE  MIRROR  COARSE   JUL 26 16:00:00  Y    UNDOTBS1.272.821807545
DATAFILE  MIRROR  COARSE   JUL 29 23:00:00  Y    UNDOTBS2.278.821809223
DATAFILE  MIRROR  COARSE   JUL 26 22:00:00  Y    UNDOTBS3.279.821809241
DATAFILE  MIRROR  COARSE   JUL 26 16:00:00  Y    USERS.271.821807537
```

The following query shows an example of how to reconstruct the ASM file hierarchy by using `V$ASM_ALIAS` in combination with other `V$ASM` views:

```
SQL> SELECT CONCAT('+'||GNAME, SYS_CONNECT_BY_PATH(ANAME, '/'))
  2    FULL_PATH, SYSTEM_CREATED, ALIAS_DIRECTORY, FILE_TYPE
  3  FROM ( SELECT B.NAME GNAME, A.PARENT_INDEX PINDEX,
  4           A.NAME ANAME, A.REFERENCE_INDEX RINDEX,
  5           A.SYSTEM_CREATED, A.ALIAS_DIRECTORY,
  6           C.TYPE FILE_TYPE
  7         FROM V$ASM_ALIAS A, V$ASM_DISKGROUP B, V$ASM_FILE C
  8         WHERE A.GROUP_NUMBER = B.GROUP_NUMBER
  9               AND A.GROUP_NUMBER = C.GROUP_NUMBER(+)
 10               AND A.FILE_NUMBER = C.FILE_NUMBER(+)
 11               AND A.FILE_INCARNATION = C.INCARNATION(+)
 12       )
 13  START WITH (MOD(PINDEX, POWER(2, 24))) = 0
 14  CONNECT BY PRIOR RINDEX = PINDEX;

FULL_PATH                                                S A FILE_TYPE
-------------------------------------------------------- - - ----------------
+DATA/cluster01                                          Y Y
+DATA/cluster01/ASMPARAMETERFILE                         Y Y
+DATA/cluster01/ASMPARAMETERFILE/REGISTRY.253.692923731 Y N ASMPARAMETERFILE
+DATA/cluster01/OCRFILE                                  Y Y
+DATA/cluster01/OCRFILE/REGISTRY.255.692923735           Y N OCRFILE
+DATA/ORCL                                               Y Y
+DATA/ORCL/DATAFILE                                      Y Y
+DATA/ORCL/DATAFILE/UNDOTBS1.258.692926341               Y N DATAFILE
+DATA/ORCL/DATAFILE/EXAMPLE.264.692926563                Y N DATAFILE
+DATA/ORCL/DATAFILE/UNDOTBS2.265.692926841               Y N DATAFILE
+DATA/ORCL/DATAFILE/USERS.259.692926341                  Y N DATAFILE
+DATA/ORCL/DATAFILE/SYSTEM.256.692926339                 Y N DATAFILE
+DATA/ORCL/DATAFILE/SYSAUX.257.692926339                 Y N DATAFILE
+DATA/ORCL/CONTROLFILE                                   Y Y
+DATA/ORCL/CONTROLFILE/Current.260.692926519             Y N CONTROLFILE
+DATA/ORCL/ONLINELOG                                     Y Y
+DATA/ORCL/ONLINELOG/group_1.261.692926525               Y N ONLINELOG
+DATA/ORCL/ONLINELOG/group_2.262.692926529               Y N ONLINELOG
+DATA/ORCL/ONLINELOG/group_4.267.692926921               Y N ONLINELOG
+DATA/ORCL/ONLINELOG/group_3.266.692926919               Y N ONLINELOG
+DATA/ORCL/TEMPFILE                                      Y Y
+DATA/ORCL/TEMPFILE/TEMP.263.692926547                   Y N TEMPFILE
+DATA/ORCL/PARAMETERFILE                                 Y Y
+DATA/ORCL/PARAMETERFILE/spfile.268.692926927            Y N PARAMETERFILE
+DATA/ORCL/spfileorcl.ora                                N N PARAMETERFILE
```

Viewing ASM Files

- In an ASM instance, `V$ASM_FILE` displays one row for every file in every disk group mounted by the instance.
- `V$ASM_FILE` is often joined with `V$ASM_ALIAS` and `V$ASM_DISKGROUP` to construct fully qualified file names.
- The ASMCMD `LS` command shows similar information:

```
ASMCMD> ls -ls DATA/ORCL/DATAFILE
Type     Redund Blk_Size Blocks      Bytes      Space Name
DATAFILE MIRROR     8192  45841 375529472  756023296 EXAMPLE.266.829668029
DATAFILE MIRROR     8192 280321 296389632 4598005760 SYSAUX.258.829667721
DATAFILE MIRROR     8192 102401 838868992 1682964480 SYSTEM.259.829667807
DATAFILE MIRROR     8192  23041 188751872  382730240 UNDOTBS1.261.829667893
DATAFILE MIRROR     8192   3201  26222592   54525952 UNDOTBS2.267.829668759
DATAFILE MIRROR     8192   3201  26222592   54525952 UNDOTBS3.268.829668765
DATAFILE MIRROR     8192    641   5251072   12582912 USERS.260.829667891
```

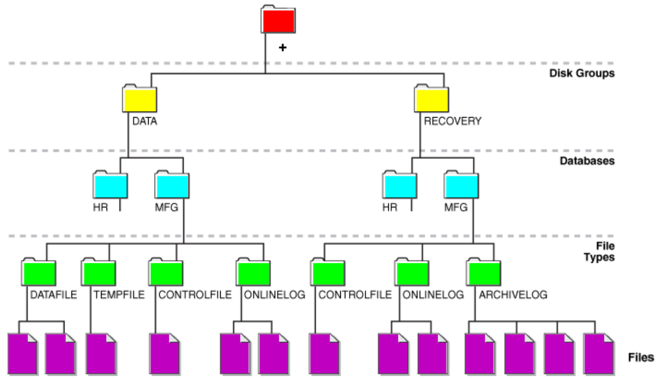
ORACLE

`V$ASM_FILE` contains information about the physical attributes of ASM files. This includes settings for striping, redundancy, and block size, along with information about the file size, type, redundancy status, creation date, and date of last modification.

Because it does not contain directory path information, `V$ASM_FILE` is often joined with `V$ASM_ALIAS` and `V$ASM_DISKGROUP` to construct fully qualified file names and full alias paths. See the description of `V$ASM_ALIAS` earlier in this lesson for an example.

ASM Directories

- ASM disk groups contain the following system-generated hierarchical directory structure for storing ASM files.



- You can create your own directories within this hierarchy to store the aliases that you create.

ASM disk groups contain a system-generated hierarchical directory structure for storing ASM files. The system-generated file name that ASM assigns to each file represents a path in this directory hierarchy. The following is an example of a system-generated file name:

```
+DATA/ORCL/DATAFILE/SYSTEM.262.676172197
```

The plus sign represents the root of the ASM file system. The `DATA` directory is the parent directory for all the files in the `DATA` disk group. The `orcl` directory is the parent directory for all the files in the `orcl` database, and the `DATAFILE` directory contains all the data files for the `orcl` database.

You can create your own directories within this hierarchy to store the aliases that you create. Thus, in addition to having user-friendly alias names for ASM files, you can have user-friendly paths to those names. For example, the following user-defined directory might be used to store a collection of alias file names:

```
+DATA/ORCL/my_directory
```

User-defined directories may be created at any level below the disk group directories. That is, you cannot create a user-defined directory at the root (+) level.

Managing ASM Directories

Use the ALTER DISKGROUP statement to create, rename, and drop ASM directories.

```
ALTER DISKGROUP diskgroup_name
  { ADD DIRECTORY 'dir_name' [, 'dir_name' ]...
  | DROP DIRECTORY
      'dir_name' [ FORCE | NOFORCE ]
      [, 'dir_name' [ FORCE | NOFORCE ]]...
  | RENAME DIRECTORY
      'old_dir_name' TO 'new_dir_name'
      [, 'old_dir_name' TO 'new_dir_name' ]...
  }
;
```

Examples:

```
SQL> ALTER DISKGROUP DATA ADD DIRECTORY '+DATA/mydir';
```

```
SQL> ALTER DISKGROUP DATA
   2 RENAME DIRECTORY '+DATA/mydir' TO '+DATA/myotherdir';
```


You can use the ALTER DISKGROUP statement to create, rename, and drop user-defined directories. You cannot rename or drop system-created directories.

When using the CREATE DIRECTORY clause, you cannot create a nested directory unless the parent directory already exists. For example, the following statement fails unless first_dir already exists:

```
ALTER DISKGROUP DATA ADD DIRECTORY '+DATA/first_dir/second_dir';
```

When using the DROP DIRECTORY clause, you cannot drop a directory that contains alias names unless you also specify the FORCE clause. If you specify DROP DIRECTORY ... FORCE, ASM recursively removes all the subdirectories and aliases that exist below the specified directory.

The ASM Command-Line utility (ASMCMD) can also be used to manage ASM directories.

The ASMCMD commands are as follows:

- mkdir: Makes a directory
- mv: Renames a file or directory
- rm: Removes a file or directory and its aliases
- rmalias: Removes the alias only

You can also perform all these operations from Enterprise Manager.

Managing Alias File Names

Use the `ALTER DISKGROUP` statement to create, rename, and drop ASM aliases.

```
ALTER DISKGROUP diskgroup_name
  { ADD ALIAS 'alias_name' FOR 'filename'
      [, 'alias_name' FOR 'filename' ]...
  | DROP ALIAS 'alias_name' [, 'alias_name' ]...
  | RENAME ALIAS 'old_alias_name' TO 'new_alias_name'
      [, 'old_alias_name' TO 'new_alias_name' ]...
  }
;
```

Examples:

```
SQL> ALTER DISKGROUP DATA ADD ALIAS '+DATA/mydir/system.dbf'
  2  FOR '+DATA/sample/datafile/system.262.676172197';
```

```
SQL> ALTER DISKGROUP DATA RENAME ALIAS '+DATA/mydir/datafile.dbf'
  2  TO '+DATA/payroll/compensation.dbf';
```

ORACLE

ASM alias file names, or aliases, provide a more user-friendly means for referring to ASM files, rather than the system-generated file names. For example, the following `ALTER DATABASE` statement uses an alias to the data file that is being taken offline:

```
ALTER DATABASE DATAFILE '+DATA/mydir/myfile.dbf' OFFLINE;
```

You can create an alias for a file when you create it in the database. For example:

```
CREATE TABLESPACE myts
    DATAFILE '+DATA/mydir/myts.dbf' SIZE 50M ONLINE;
```

The alias name is created automatically, but the command fails if the directory alias does not already exist.

Note: A file that is created thus is not an OMF file.

You can add an alias to an existing file by using the `ADD ALIAS` clause of the `ALTER DISKGROUP` statement. You can create an alias in any system-generated or user-created ASM directory that is already in existence. However, you cannot create an alias at the root level (+). You cannot create multiple aliases for the same ASM file.

The `ALTER DISKGROUP` statement also provides clauses to `RENAME` and `DROP` existing aliases. Dropping an alias does not drop the referenced file. If you use the `ALTER DISKGROUP ... DROP FILE` statement to drop an ASM file, the alias associated with the file is also removed.

ASMCMD can also be used to manage ASM aliases.

Disk Group Templates

- Templates are used to set striping and redundancy attributes for ASM files.
- The striping attribute options are:
  - `FINE`: 128 KB stripe size
  - `COARSE`: 1 AU stripe size
- The redundancy attribute options are:
  - `MIRROR`: Two-way mirroring
  - `HIGH`: Three-way mirroring
  - `UNPROTECTED`: No ASM mirroring
- A set of default templates is created for each disk group.
  - Default template settings depend on disk group redundancy.
- The redundancy attribute applies only to `NORMAL` redundancy disk groups.

ORACLE

Templates are used to set the redundancy (mirroring) and striping attributes of each individual file that is created in an ASM disk group. When a file is created, redundancy and striping attributes are set for that file based on an explicitly named template or the system template that is the default template for the file type.

When a disk group is created, ASM creates a set of default templates for that disk group. The set consists of one template for each file type (data file, control file, redo log file, and so on) that is supported by ASM. For example, a template named `ONLINELOG` provides the default file redundancy and striping attributes for all redo log files that are written to that disk group. The default template settings for redundancy depend on the disk group type. The default template for data files for a normal redundancy disk group sets two-way mirroring, whereas the corresponding default template in a high redundancy disk group sets three-way mirroring. You can modify the default templates.

The striping attribute of templates applies to all disk group types (normal redundancy, high redundancy, and external redundancy). In practice, the redundancy attribute of templates applies only to normal redundancy disk groups. It is effectively ignored for high redundancy disk groups in which every file is always three-way mirrored. It is also effectively ignored for external redundancy disk groups in which no files are mirrored by ASM. Nevertheless, each type of disk group gets a full set of templates, and the redundancy value in each template is always set to the proper default for the disk group type.

The following table lists the default templates and the attributes that are associated with the different ASM file types. As the table shows, the initial redundancy value of each default template depends on the type of disk group that the template belongs to.

Template Name (File Type)	**Striping**	**Mirroring, Normal Redundancy Disk Group**	**Mirroring, High Redundancy Disk Group**	**Mirroring, External Redundancy Disk Group**
CONTROLFILE	FINE	HIGH	HIGH	UNPROTECTED
DATAFILE	COARSE	MIRROR	HIGH	UNPROTECTED
ONLINELOG	COARSE	MIRROR	HIGH	UNPROTECTED
ARCHIVELOG	COARSE	MIRROR	HIGH	UNPROTECTED
TEMPFILE	COARSE	MIRROR	HIGH	UNPROTECTED
BACKUPSET	COARSE	MIRROR	HIGH	UNPROTECTED
PARAMETERFILE	COARSE	MIRROR	HIGH	UNPROTECTED
DATAGUARDCONFIG	COARSE	MIRROR	HIGH	UNPROTECTED
FLASHBACK	COARSE	MIRROR	HIGH	UNPROTECTED
CHANGETRACKING	COARSE	MIRROR	HIGH	UNPROTECTED
DUMPSET	COARSE	MIRROR	HIGH	UNPROTECTED
XTRANSPORT	COARSE	MIRROR	HIGH	UNPROTECTED
AUTOBACKUP	COARSE	MIRROR	HIGH	UNPROTECTED
OCRFILE	COARSE	MIRROR	HIGH	UNPROTECTED
ASMPARAMETERFILE	COARSE	MIRROR	HIGH	UNPROTECTED

Viewing Templates

- In an ASM or a database instance, V$ASM_TEMPLATE displays one row for every template present in every disk group that is mounted by the ASM instance.
- The ASMCMD lstmpl command shows all the templates for a given disk group.

```
$ asmcmd lstmpl -l -G data
Group Group
Name  Num  Name              Stripe  Sys Redund PriReg  MirrReg
DATA  1    ARCHIVELOG        COARSE  Y   MIRROR COLD    COLD
DATA  1    ASMPARAMETERFILE  COARSE  Y   MIRROR COLD    COLD
DATA  1    AUDIT_SPILLFILES  COARSE  Y   MIRROR COLD    COLD
DATA  1    AUTOBACKUP        COARSE  Y   MIRROR COLD    COLD
DATA  1    AUTOLOGIN_KEY_STORE COARSE Y  MIRROR COLD    COLD
DATA  1    BACKUPSET         COARSE  Y   MIRROR COLD    COLD
DATA  1    CHANGETRACKING    COARSE  Y   MIRROR COLD    COLD
DATA  1    CONTROLFILE       FINE    Y   HIGH   COLD    COLD
DATA  1    DATAFILE          COARSE  Y   MIRROR COLD    COLD
...
```

ORACLE

The V$ASM_TEMPLATE view displays disk group templates and their attribute settings. The following query lists all the templates that are associated with the DATA disk group. The SYSTEM column returns Y for system-generated templates and N for user-defined templates.

```
SQL> SELECT T.NME, T.REDUNDANCY, T.STRIPE, T.SYSTEM
  2  FROM V$ASM_TEMPLATE T, V$ASM_DISKGROUP G
  3  WHERE T.GROUP_NUMBER = G.GROUP_NUMBER
  4  AND G.NAME = 'DATA';

NAME                           REDUND STRIPE S
------------------------------ ------ ------ -
PARAMETERFILE                  MIRROR COARSE Y
ASMPARAMETERFILE               MIRROR COARSE Y
OCRFILE                        MIRROR COARSE Y
DATAGUARDCONFIG                MIRROR COARSE Y
AUDIT_SPILLFILES               MIRROR COARSE Y
AUTOLOGIN_KEY_STORE            MIRROR COARSE Y
KEY_STORE                      MIRROR COARSE Y
FLASHBACK                      MIRROR COARSE Y
CHANGETRACKING                 MIRROR COARSE Y
XTRANSPORT                     MIRROR COARSE Y
AUTOBACKUP                     MIRROR COARSE Y
...
```

Managing Disk Group Templates

Use the ALTER DISKGROUP statement to add, modify, and drop disk group templates.

```
ALTER DISKGROUP diskgroup_name
  { { ADD | MODIFY } TEMPLATE
      qualified_template_clause [, qualified_template_clause ]...
  | DROP TEMPLATE template_name [, template_name ]...
  }
;

qualified_template_clause ::= template_name ATTRIBUTES
  ([ MIRROR | HIGH | UNPROTECTED ] [ FINE | COARSE ])
```

Examples:

```
SQL> ALTER DISKGROUP DATA ADD TEMPLATE unprot ATTRIBUTES (UNPROTECTED);
```

```
SQL> ALTER DISKGROUP DATA ADD TEMPLATE reliable ATTRIBUTES (HIGH FINE);
```

```
SQL> ALTER DISKGROUP DATA MODIFY TEMPLATE DATAFILE ATTRIBUTES (FINE);
```

ORACLE

Creating your own templates allows you to set the right combination of attributes to meet your requirements. It also allows you to specify an intuitive name for the attribute set.

For example, in a normal or high redundancy disk group, you might use a template with a redundancy setting of UNPROTECTED to create transient files without the overhead of mirroring. This can be a useful way of creating short-term databases for testing and development purposes.

Using the ALTER DISKGROUP statement, you can add new templates to a disk group, modify existing ones, or drop templates.

When adding a new template, you can omit either the redundancy or the stripe attribute setting. If you omit an attribute setting, it defaults as follows:

- For the redundancy attribute, the value defaults to MIRROR for a normal redundancy disk group, HIGH for a high redundancy disk group, and UNPROTECTED for an external redundancy disk group.
- For the stripe attribute, the value defaults to COARSE.

Managing Disk Group Templates with ASMCMD

- Use the mktmpl statement to add disk group templates.

```
ASMCMD> mktmpl -G DATA --redundancy unprotected unprotect;
```

- Use the chtmpl command to modify disk group templates.

```
ASMCMD> chtmpl –G DATA --striping fine DATAFILE;
```

- Use the rmtmpl command to drop disk group templates.

```
ASMCMD> rmtmpl –G DATA unprotect;
```

- Use the lstmpl command to view disk group templates.

```
ASMCMD> lstmpl –G DATA
```


You can modify both user-defined and system-generated default templates. Modifying a template does not change the attributes of any existing files. The attributes of a modified template are applied only to new files that reference the modified template, either explicitly or implicitly. You can drop only user-defined templates, not system-default templates.

The V$ASM_TEMPLATE view lists all the templates known to the ASM instance.

Using Disk Group Templates

You can apply disk group templates to newly created files in the following ways:

- Alias file name with template

```
SQL> ALTER TABLESPACE myspace ADD
  2 DATAFILE '+DATA(mytemplate)/mydir/myspace02.dbf' SIZE 50M;
```

- Incomplete file name with template

```
SQL> ALTER TABLESPACE myspace ADD
  2 DATAFILE '+DATA(mytemplate)' SIZE 50M;
```

- `DB_CREATE_FILE_DEST` database initialization parameter

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '+DATA(mytemplate)';
SQL> CREATE TABLESPACE yourspace;
```

ORACLE

You can reference a template name when creating a file by using either an alias or an incomplete file name. This allows you to assign desired attributes based on an individual file rather than the file type.

You can also use an incomplete file name with template in the `DB_CREATE_FILE_DEST` database initialization parameter. Using this setting applies the named template to files created by statements such as `CREATE TABLESPACE` and `ALTER TABLESPACE`.

ASM Access Control Lists

ASM Access Control Lists (ACLs):

- Set permissions at the ASM file level
- Set grants to groups or users
  - Users are Database software owners.
  - Users are identified by the OS user ID.
- The ASM ACL includes no passwords.
  - ASM trusts the OS authentication mechanisms.

ASM access control lists (ACLs) provide optional protection for the ASM files. The objective of the ASM access control list is not security but separation of duties to prevent accidental file damage. Without ACLs, any user with `SYSDBA` privilege may access the ASM files in the mounted disk group, up to and including removing them.

To set up Oracle ASM File Access Control, you must create separate OS groups for the OSASM, OSDBA for ASM, and OSDBA for database groups. The OSDBA group for the database must be different for each database instance that uses the same ASM instance.

Each ASM file is created by a `DBUSER`. This `DBUSER` is usually an owner of a database instance. The ASM files that are created for that database are owned by that `DBUSER`. The operating system user ID of the database software owner identifies the `DBUSER` to the ASM instance. Access is limited by the operating system effective user ID `NUMBER` of the `DBUSER`. The operating system user of a running database instance is automatically added to a disk group when the database instance accesses that disk group and creates files.

Each `DBUSER` can create access control lists.

The ASM ACL includes a user group, a list of `DBUSERS`, but not any passwords. ASM trusts the OS authentication mechanisms. A `DBUSER` is a member of the OSDBA group for ASM.

ASM File Access Control Available on Windows

- ASM file access control enables file ownership and permission settings on ASM files.
  - Conceptually similar to UNIX file ownership and permissions
- In earlier versions:
  - ASM file access control was infeasible on Windows because Oracle had to run by using the `LOCALSYSTEM` account.
- With Oracle Database 12*c*:
  - Any Windows user can install and run Oracle.
  - ASM file access control can be implemented.
    - Functionally equivalent to UNIX and Linux platforms

ORACLE

ASM file access control restricts file access to specific Oracle ASM clients that connect to ASM. An ASM client is typically a database, which is identified by the user that owns the database home directory. ASM file access control is particularly useful in environments that are used to consolidate many Oracle databases, and can be used to ensure that database administrators can access only those database files that are associated with the databases under their control.

In earlier versions, ASM file access control could not be implemented on Windows because ASM and database processes (threads) had to run by using the `LOCALSYSTEM` account.

Oracle Database 12*c* removes this restriction and enables different Windows users to install and run Oracle Database and ASM. This, in turn, allows ASM file access control to be made available on Windows, with functionality equivalent to UNIX and Linux platforms.

ASM ACL Prerequisites

Access control lists for ASM files require:

- Linux or UNIX operating system
- Job role separation at the OS level
- Diskgroup attributes:
  - `COMPATIBLE.ASM` to 11.2 or later
  - `COMPATIBLE.RDBMS` to 11.2 or later
  - `ACCESS_CONTROL.ENABLED` to `TRUE`
  - `ACCESS_CONTROL.UMASK` to a mask value
- Permissions: none (0), read (4), or read-write(6)
- Mask values: 6 (removes all), 2 (removes write), 0 (removes nothing)

ORACLE

Access control lists for ASM are available only on Linux and UNIX. Job role separation must be configured. Each database and ASM instance must have different owners.

Disk group attributes must be set. `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` must be set to 11.2 or later. For each disk group that is using ACLs, set `ACCESS_CONTROL.ENABLED` to `TRUE` and set `ACCESS_CONTROL.UMASK` to a mask value. `ACCESS_CONTROL.ENABLED` must be set to `TRUE` before `ACCESS_CONTROL.UMASK` can be set. The umask value removes permissions from full access of read-write for owner, group, and others. In permissions, 6 indicates read-write, 4 indicates read, and 0 indicates none. A umask value of 0 removes nothing; 2 removes the write privilege; 6 removes the read-write privilege. Concatenating the values gives permissions for a user, group, or others. For example, a umask of 026 will cause files to have permissions of 640, which is read-write for the owner, read for the group, and no access for all other users.

```
ALTER DISKGROUP DATA2 SET ATTRIBUTE 'access_control.enabled' = 'true';
```

```
ALTER DISKGROUP DATA2 SET ATTRIBUTE 'access_control.umask' = '026';
```

Equivalent ASMCMD commands:

```
ASMCMD> setattr -G data2 access_control_enabled true
```

```
ASMCMD> setattr -G data2 access_control_umask 026
```

Attributes of the disk group can also be set with the `asmca` tool.

Managing ASM ACL with SQL Commands

The ACL for ASM can be managed with SQL commands.

```
ALTER DISKGROUP ADD USERGROUP ... WITH MEMBER
ALTER DISKGROUP DROP USERGROUP
ALTER DISKGROUP MODIFY USERGROUP ADD MEMBER
ALTER DISKGROUP MODIFY USERGROUP DROP MEMBER
ALTER DISKGROUP ADD USER
ALTER DISKGROUP DROP USER
ALTER DISKGROUP SET PERMISSION
ALTER DISKGROUP SET OWNERSHIP
SELECT * FROM V$ASM_USER
SELECT * from V$ASM_USERGROUP
```


The access control lists in ASM can be managed with the SQL commands shown in the slide.

To create a user group ACL, use the following command:

```
ALTER DISKGROUP ADD USERGROUP groupname WITH MEMBER user [, user]
```

Each user must already exist.

Any user with the `SYSDBA` or `SYSASM` privilege may create groups. Only the group owner or an ASM administrator may change or drop groups. Only an ASM administrator may add or drop users. Only the owner of a file or an ASM administrator may change the ownership of a file.

Users of files are usually the database owners, and users are added automatically as files are created in a disk group. Adding users to a disk group should seldom be needed. An OS user that is not a database owner could be added to a disk group and a user group.

The `SET PERMISSION` clause modifies the permissions of an Oracle ASM file. Note that setting read-only permission to a file that has read-write permission revokes the write permission. Only the file owner or the Oracle ASM administrator can change the permissions of a file. You cannot change permissions on an open file.

You cannot change ownership on an open file.

Managing ASM ACL with ASMCMD Commands

The ACL for ASM can be managed with ASMCMD commands:

```
chgrp usergroup list_of_files
chmod mode list_of_files
chown user[:usergroup ] list_of_files
groups diskgroup user
grpmod { --add | --delete } diskgroup usergroup user1 [user2]…
lsgrp [-Ha] [ -G diskgroup ] [ pattern_expr ]
lsusr [-Ha] [ -G diskgroup ] [ pattern_expr ]
mkgrp diskgroup usergroup [user1] [user2]...
mkusr diskgroup user
passwd user
rmgrp diskgroup usergroup
rmusr diskgroup user
```

ORACLE

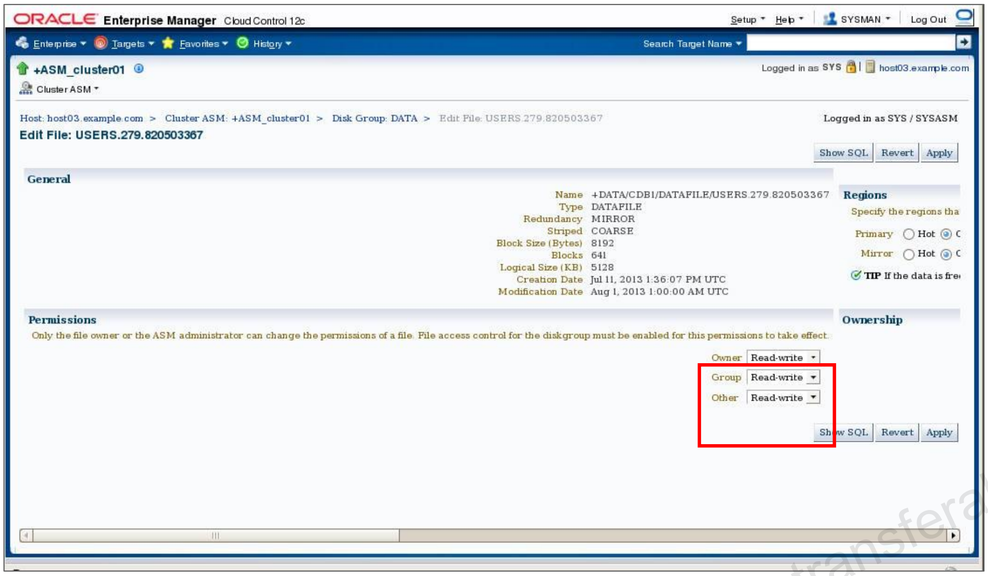
The `ASMCMD` commands allow the storage administrator to perform the same operations as were shown with SQL commands but in a more familiar format. The `asmcmd` commands are patterned after the UNIX commands. The `asmcmd` commands have the same restrictions as the SQL commands.

- `chmod`: Modifies the permissions of an Oracle ASM file or list of files. Note that setting read-only permission to a file that has read-write permission revokes the write permission. Only the file owner or the Oracle ASM administrator can change the permissions of a file. You cannot change permissions on an open file.
- `chown`: You cannot change ownership on an open file.

In ASMCMD, a user can connect by using the `–a` option to set privileges, either `SYSASM` or `SYSDBA`, but the OS user that is invoking the command must be a member of the OS group that has that privilege. `SYSASM` is the default. To connect with the `SYSDBA` privilege, use the following syntax:

```
asmcmd --privilege sysdba
```

Managing ASM ACL with Enterprise Manager



The ASM access control lists can be viewed and managed through Enterprise Manager. On the Edit File page, the permissions and ownership of an ASM file can be modified. Ownership and permissions cannot be changed on an open file.

ASM ACL Guidelines

- Use ACLs to prevent accidental file damage.
- Be careful when enabling access control if it blocks a user's access to a critical file.

ORACLE

ASM access control lists are intended as a safety feature rather than a security feature. Without access control lists, any user with `SYSDBA` privileges can manipulate any file in the disk group, possibly unintentionally dropping or damaging files belonging to other databases. This becomes more important with storage consolidation where multiple databases share a single ASM installation, either on a single server or a clustered one.

To implement ACLs, each database must have a different OS owner and group, and the ASM instance must have yet another owner as shown in the lesson titled "Oracle Grid Infrastructure Architecture." Then even if the databases share disk groups, ACLs can prevent a user with the `SYSDBA` privilege in ASM from damaging a file belonging to another database.

When configuring ACL permissions and changing ownership, be careful that the ACL does not block access to a critical file or disk group. For example, make sure that the database can access the disk group that contains the control file.

Quiz

RMAN is the only way to back up an Oracle Database that resides in ASM.

a. True

b. False

Answer: b

It is possible to back up an Oracle Database by using a custom PL/SQL routine with `DBMS_FILE_TRANSFER` or by using the XML DB repository. RMAN is, however, the preferred method of backing up Oracle Databases that reside in ASM. RMAN is very well integrated with ASM and provides many advanced backup and recovery features to aid administrators.

Quiz

Which two ASM file name forms will always be a valid way of referencing an existing ASM file?

a. Fully qualified
b. Numeric
c. Alias
d. Alias with template
e. Incomplete
f. Incomplete with template

ORACLE

Answer: a, b

Option c is not the correct answer because there might not be an alias defined for some ASM files.

Summary

In this lesson, you should have learned how to:

- Use different client tools to access ASM files
- Describe the format of a fully qualified ASM file name
- Explain how ASM files, directories, and aliases are created and managed
- Describe and manage disk group templates

Practice 7 Overview: Administering ASM Files, Directories, and Templates

This practice covers the following topics:

- This practice covers managing ASM files, directories, and templates to control the file level redundancy in a disk group.

8

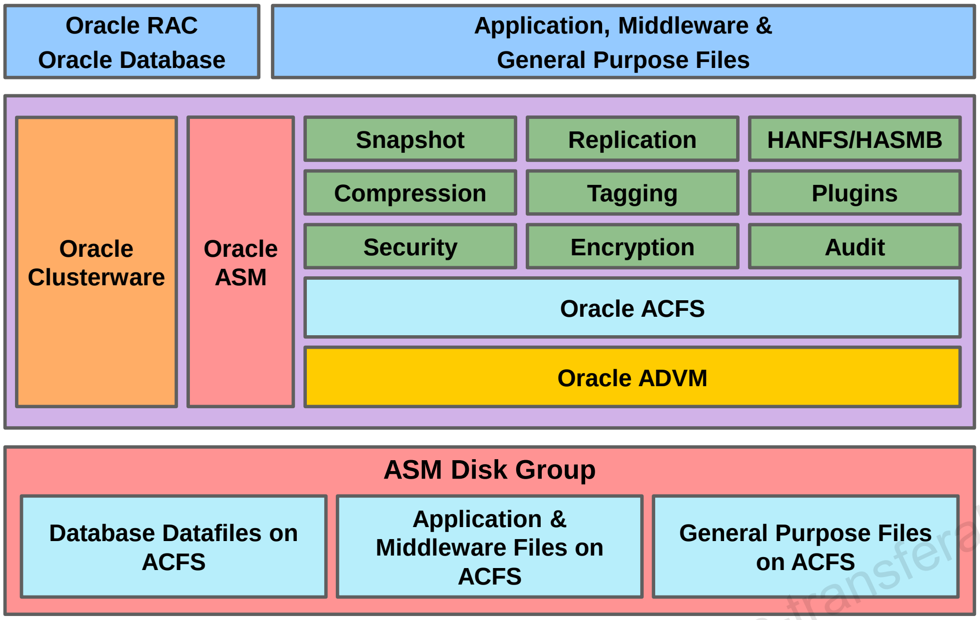
Administering Oracle ACFS

Objectives

After completing this lesson, you should be able to:

- Administer ASM Dynamic Volume Manager
- Manage ASM volumes
- Implement ASM Cluster File System
- Manage ASM Cluster File System (ACFS)
- Use ACFS snapshots

Oracle ACFS



Oracle ACFS is a multi-platform, scalable file system, and storage management technology that extends Oracle ASM functionality to support all customer files. Oracle ACFS supports Oracle Database files and application files, including executables, database data files, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and other general-purpose application file data. Oracle ACFS conforms to POSIX standards for Linux and UNIX, and to Windows standards for Windows.

An Oracle ACFS file system communicates with Oracle ASM and is configured with Oracle ASM storage, as shown in the slide. Oracle ACFS leverages Oracle ASM functionality that enables:

- Oracle ACFS dynamic file system resizing
- Maximized performance through direct access to Oracle ASM disk group storage
- Balanced distribution of Oracle ACFS across Oracle ASM disk group storage for increased I/O parallelism
- Data reliability through Oracle ASM mirroring protection mechanisms

Oracle ACFS is tightly coupled with the Oracle Clusterware technology, participating directly in Clusterware cluster membership state transitions and in Oracle Clusterware resource-based high availability (HA) management. In addition, the Oracle installation, configuration, verification, and management tools have been updated to support Oracle ACFS.

Oracle ACFS included the following features in Oracle ASM 12*c* Release 1:

- High availability NFS
- Support for all Oracle Database files
- Snapshot enhancements
- Advanced auditing
- Metrics plug-in
- Replication enhancements
- Tagging API
- Resource enhancements

Some of these features are discussed in this lesson and Appendix A.

The following are changes to Oracle ACFS and Oracle ASM ADVM for Oracle ASM 12*c* Release 2.

- Oracle ACFS Snapshot-Based Replication
- Oracle ACFS Snapshot Enhancements
- Oracle ACFS Compression
- Oracle ACFS Defragger
- Oracle ACFS Support for 4K Sectors
- Oracle ACFS Automatic Resize
- Oracle ACFS Metadata Acceleration
- Oracle ACFS NAS Maximum Availability eXtensions
- Oracle ACFS Plugins for File Content Data Collection
- Oracle ACFS Sparse Files
- Oracle ACFS Scrubbing Functionality
- Oracle ACFS Loopback Functionality
- Oracle ACFS Diagnostic Commands

Some of these features are discussed in lesson 9 and Appendix A.

Enhanced Platform Support for ACFS Data Services

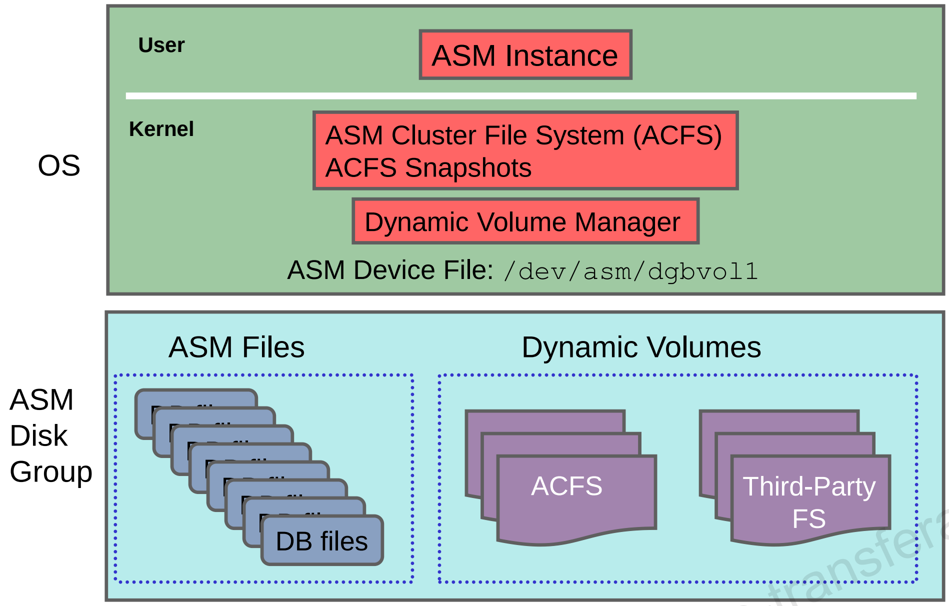
Feature \ OS	Linux	Windows	Solaris	AIX
Read Only Snapshots	11.2.0.1	11.2.0.1	11.2.0.2	11.2.0.2
ReadWrite Snapshots	11.2.0.3	11.2.0.3	11.2.0.3	11.2.0.3
Tagging	11.2.0.2	11.2.0.3	12.1.0.1	12.1.0.1
Replication	11.2.0.2	11.2.0.3	12.1.0.1	12.1.0.1
Encryption	11.2.0.2	11.2.0.3	12.1.0.1	-
Enhanced Security	11.2.0.2	11.2.0.3	12.1.0.1	-
General Purpose Files	11.2.0.1	11.2.0.1	11.2.0.2	11.2.0.2
RMAN, Datapump, ArchiveLog, FRA	11.2.0.3	11.2.0.3	11.2.0.3	11.2.0.3
Database files (Datafiles, Controlfiles, Redologs, etc.)	12.1.0.1	12.1.0.2	12.1.0.1	12.1.0.1
Accelerator Volume	12.2.0.1	-	-	-
Compression	12.2.0.1	-	-	-

ORACLE

The table in the slide summarizes the platform support for the different ACFS data service capabilities. The platform-specific enhancements for Oracle Database 12*c* are highlighted.

For more information about ACFS Support, refer to *MOS note: Doc ID 1369107.1 (ACFS Support On OS Platforms (Certification Matrix)*

ASM Files and Volumes



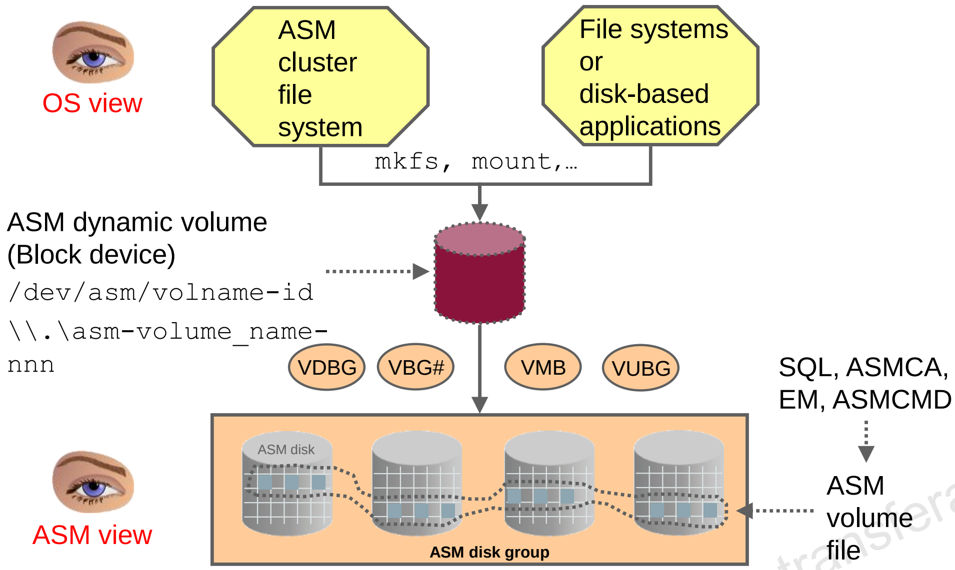
Oracle ASM includes support for a general-purpose cluster file system, the ASM Cluster File System (ACFS). At the operating system (OS) level, the ASM instance provides the disk group, which is a logical container for physical disk space. The disk group can hold ASM database files and ASM dynamic volume files. The ASM Dynamic Volume Manager (ADVM) presents the volume device file to the operating system as a block device. The `mkfs` utility can be used to create an ASM file system in the volume device file.

Three OS kernel modules that are loaded in the OS provide the data service. On Linux, they are: `oracleadvm`, the ASM Dynamic Volume Manager module; `oracleoks`, the kernel services module; and `oracleacfs`, the ASM file system module.

```
# lsmod|grep oracle
oracleacfs            3053229  2
oracleadvm             320180  8
oracleoks              417171  2 oracleacfs,oracleadvm
```

These modules provide the ASM Cluster File System, ACFS snapshots, ADVM, and cluster services. The ASM volumes are presented to the OS as a device file at `/dev/asm/`<*volume name*>-<*number*>. The volume device file appears as another ASM file to the ASM instance and `asmcmd` utility. The ASM layers are transparent to the OS file system commands. Only the files and directories that are created in ACFS and the ACFS snapshots are visible to the OS file system commands. Other file system types such as ext4 and NTFS may be created in an ADVM volume by using the `mkfs` command on Linux and `advmutil` commands on Windows.

ACFS and ADVM Architecture: Overview



ADVM provides volume management services and a standard disk device driver interface to clients. Clients, such as file systems and other disk-based applications, issue I/O requests to ADVM volume devices as they would to other storage devices on a vendor operating system.

ADVM extends ASM by providing a disk driver interface to storage backed by an ASM file. The administrator can use the ADVM to create volumes that contain file systems. These file systems can be used to support files beyond Oracle database files such as executables, report files, trace files, alert logs, and other application data files. With the addition of ADVM and ACFS, ASM becomes a complete storage solution of user data for both database and non-database file needs. ACFS is intended as a general file system that is accessible by the standard OS utilities. ACFS can be used in either a single server or a cluster environment.

ASM volumes serve as containers for storage presented as a block device that is accessed through ADVM. File systems or user processes can perform I/O on this “ASM volume device” just as they would on any other device. To accomplish this, ADVM is configured into the operating system. A volume device is constructed from an ASM file. ASM file extents map the ASM volume file to logical blocks located on specific physical devices. Additional processes are started as part of the ASM instance and serve as intermediaries between the ASM instance and ADVM. To use the ADVM driver, an ASM instance must exist with at least one disk group mounted that can be used to contain an ASM volume file.

An ASM volume is an ASM file. It inherits the properties of the ASM disk group and behaves similar to any other ASM file. ASM volume storage is automatically rebalanced whenever a storage configuration change occurs. This reconfiguration can be performed while an ASM volume is in use. Because ASM uses direct I/O, ASM volumes offer performance equivalent to raw disks.

An OS device file is created automatically when an ASM volume is created by using either `asmcmd`, SQL*Plus, ASMCA, or the Enterprise Manager graphical interfaces. On Linux, this device file is created in the `/dev/asm` directory. You can configure both disk group mount and volume-enable operations to occur automatically on ASM instance startup. The volume device file names are unique clusterwide and persistent across all the nodes in the cluster that have an ASM instance running with the disk group mounted and volumes enabled.

On Linux system startup, the Oracle clusterware startup loads the drivers (`oraclesacfs`, `oracleoks`, and `oracleadvm`). The ASM instance is started by the ASM cluster registry service (CRS) agent, which will also mount the appropriate ASM disk groups and enable volumes. The CRS agent then mounts any ACFS file systems in the Oracle Cluster Registry (OCR).

Similar actions are performed on Windows.

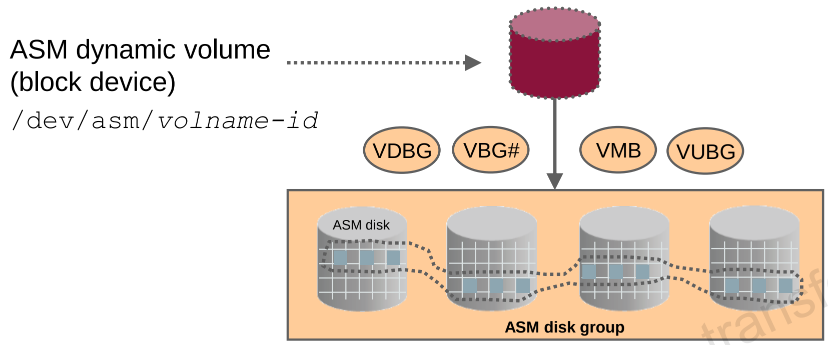
ACFS file systems are accessed through OS file system tools and APIs on UNIX and Linux systems, and accessed through Windows file system tools and APIs on Windows systems. Remote access is supported by using standard NAS file access protocols such as network file systems (NFS) and common Internet file system (CIFS) in support of heterogeneous file data sharing.

The ACFS File System and ADVM components are installed onto each host along with the other ASM components into the Grid Infrastructure home location. The ACFS components consist of drivers that are dynamically loadable OS modules, several command-line tools, and a set of processes that execute within the ASM instance. However, loading the ACFS drivers requires `root` privileges on UNIX and Linux and Administrator privileges on Windows. So, the configuration and loading of the ACFS drivers is performed by the `root` scripts that are associated with the Oracle Grid Infrastructure installation.

ACFS file systems are generally mounted on all cluster synchronization service (CSS) cluster members. In the event of a member failure, another cluster member recovers any outstanding metadata transactions on behalf of the failed member. In addition, any lock tokens that are held by the failed cluster members are recovered and the failed member is I/O fenced from the active CSS cluster. Following recovery, access by other active cluster members and any remote client systems may resume.

ADVM and ACFS Processes

- ACFS background process (ACFS)
- Volume Driver Background (VDBG)
- Volume Background (VBG*n*)
- Volume Membership Background (VMB)
- Volume Umbilicus Background (VUBG)



A small number of processes will be added to the ASM instance when a volume is enabled. These processes are not started when there are no volumes configured.

- The Volume Driver Background (VDBG) process forwards ASM requests to lock or unlock an extent for rebalancing, to resize the volume, to offline the disk, to add or drop a disk and to force and dismount a disk group to the volume manager driver. The VDBG is a fatal background process; so terminating this process brings down the ASM instance.
- The Volume Background (VBG*n*) processes wait for requests from the ADVM driver that need to be coordinated with the ASM instance. An example of such a request would be the opening or closing of an ASM volume file when the dynamic volume manager driver receives an open for a volume (possibly due to a file system mount request) or close for an open volume (possibly due to a file system unmount request), respectively. The unplanned death of any of these processes doesn't have an effect on the ASM instance.
- The Volume Membership Background (VMB) process performs the role of an IO barrier/IO fencing function. In the event of an ASM instance failure, this process continues to exist until the ADVM driver has had a chance to write out pending IOs.
- The Volume Umbilicus Background (VUBG) process relays messages between Oracle ASM instance and Oracle ASM Proxy instance used by ADVM (for ACFS).
- The ACFS background process (ACFS) within ASM manages all of the clusterware membership and state transitions.

Oracle ACFS

- ACFS is a general-purpose, single-node, and clusterwide file system that delivers support for all customer files.
- ACFS can be managed by using:
  - Native OS file system APIs and command-line tools
  - ASMCA
- ACFS supports large files with 64-bit file and file system data structure size.
- Variable extent–based storage allocation and high-performance directories contribute to fast performance.
- File system integrity and fast recovery are achieved with ACFS metadata checksums and journaling.

ORACLE

Oracle ACFS is a general-purpose, single-node, and clusterwide file system that delivers support for all customer files. Users and applications can access and manage ACFS by using native operating system file system APIs and command-line interface tools. Users can also manage Oracle ACFS with ASMCA.

ACFS supports large files with 64-bit file and file system data structure sizes, leading to exabyte-capable file and file system capacities on 64-bit platforms. Variable extent–based storage allocation and high-performance directories contribute to fast performance and shared disk configurations that provide direct storage paths to ACFS file data from each cluster member. File system integrity and fast recovery are achieved with ACFS metadata checksums and journaling.

ACFS is designed as a multi-node, shared file system model that delivers coherent, cached, and direct storage paths to ACFS file data from each cluster member. Oracle ACFS file systems are typically configured for clusterwide access. File systems, files, and directories are accessible from all cluster nodes and can be referenced by users and applications by using the same path names from any cluster node. This design enables simplified application deployments across cluster nodes and facilitates both multiple instance cluster applications and high availability (HA) failover of unmodified single-node applications.

Oracle ACFS presents single system file access semantics across cluster configurations. Applications and users on all cluster members are always presented with the same view of the shared Oracle ACFS file data, supported by the Oracle ACFS clusterwide user and metadata cache coherency mechanism.

Space Allocation

A volume allocates space in Volume Allocation Units (VAU) at creation and resize.

- VAU is stripe column multiplied by volume extent size.
- Volume extent is statically assigned based on disk group AU.
  - Volume extent is 64 MB when the AU is 1 MB.
  - Each extent is allocated round-robin on different disks.
- Stripe column is the number of stripes used inside a volume.
  - Stripe column can range from 1 to 8 (12.0.0.2 default is 8).

ORACLE

ADVM allocates space from an ASM disk group for a volume. A volume allocation unit is the smallest allocation when a volume is created and the smallest amount of space that can be added to a volume.

The VAU is based on the volume extent and the value of the stripe column. The volume extent is assigned based on the disk group allocation unit (AU). The AU can vary from 1 MB to 64 MB in powers of 2 (2, 4, 8,…, 64). The volume extent is assigned based on the AU. The default AU of 1 MB has a volume extent size of 64 MB.

The value of the stripe column is the number of volume extents allocated in each VAU. Thus VAU is the stripe column multiplied by the volume extent. The volume extents are allocated round-robin (each on the next disk in the disk group). If the stripe column is set to 1, volume striping is turned off.

Example: If the stripe column is 4 and the AU is 1 MB, the volume extent is 64 MB and the VAU is 256 MB. When the volume is created, multiples of the VAU are allocated. If you asked for a volume of 300 MB in size, you would get a volume of 512 MB, assuming that the space is available in the disk group. If you resized the volume, the space would be added or dropped in 256 MB chunks.

Striping Inside the Volume

Inside a volume, space is allocated based on the following:

- Stripe column: The number of stripes (default 8)
- Stripe width: The size of each stripe (default 1 MB)

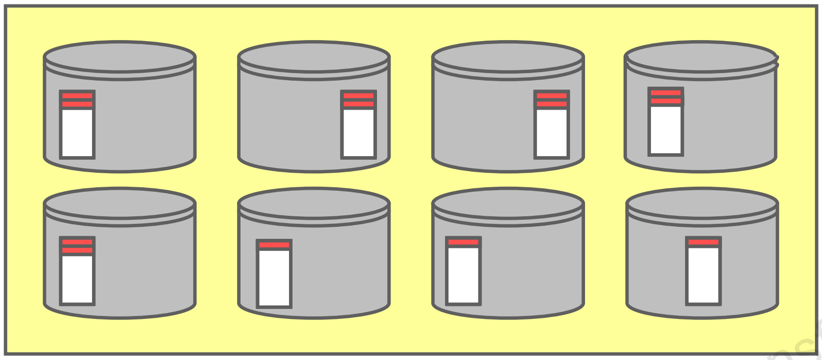
Stripe column and stripe width are properties that are set when you create the volume. The stripe column is used both for allocating initial space for the volume and also for the way the space inside the volume is used.

Volume is formatted for a file system. Volumes created while running 12.1.0.2 or later default to the high performance configuration (stripe columns = 8 and stripe width = 1 MB). Volumes created before 12.1.0.2 default to stripe columns =1 and stripe width = 128 KB. As the file system allocates space for files, the files are stored in pieces that are the size of the stripe width. Each piece goes into the next volume extent. Because the volume extents are placed on disks in the disk group in the same manner, as the pieces are placed in the extents, the pieces are spread across the disks in the disk group.

Volume Striping: Example

Example:

- Stripe column is 8; stripe width is 16 KB.
- The first 200 KB of a file inside the file system is written as 13 chunks of 16 KB stripes across eight allocation units.



For this example, assume the default AU size of 1 MB. This means that the volume extent is 64 MB. For simplicity, assume that there are eight disks in the disk group.

Because the stripe column is 8, eight volume extents are created in every volume allocation unit of 512 MB. Eight volume extents are shown, one on each of the eight disks.

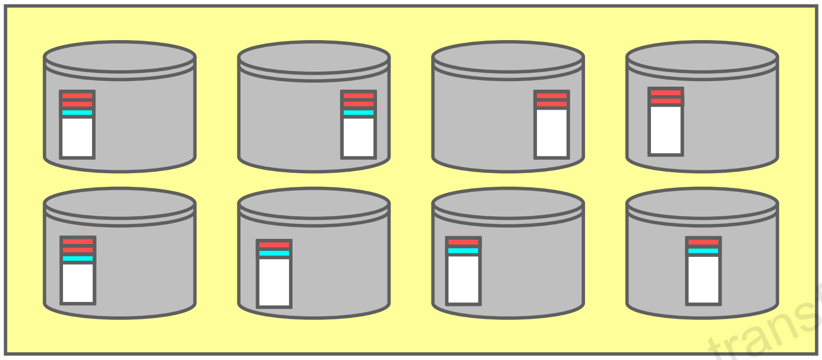
Note: Each VAU is filled before the next VAU is used.

The diagram in the slide shows how ACFS volume striping works. The strip width is 16 KB. A new file of 200 KB is written, occupying 13 chunks of 16 KB spread round-robin across the eight volume extents. Consequently, reads and writes to this file are spread across eight disks instead of one.

Volume Striping: Example

Example:

- Disk group with eight disks and external redundancy
- Default AU size of 1 MB in use
- Another ACFS file of 90 KB written as 6 chunks of 16 KB stripes across the same eight allocation units



Continuing the previous example, the next file of 90 KB is spread across each of the same volume extents in 16 KB chunks. This pattern continues until the first set of volume allocation units is filled and another set is allocated.

Creating an ACFS Volume

- Create the volume:

```
$ asmcmd volcreate -G DATA -s 100M testvol
```

- View the volume information:

```
$ asmcmd volinfo –G DATA  testvol
```

- Make a mount point directory:

```
$ mkdir /u01/app/oracle/acfsdata/testvol
```

- Make the file system (as root):

```
# mkfs -t acfs /dev/asm/testvol-403
```

- Mount the file system to the mount point:

```
# mount -t acfs /dev/asm/testvol-403 \
  /u01/app/oracle/acfsdata/testvol
```

- Register the volume:

```
$ /sbin/acfsutil registry -a -f /dev/asm/testvol-403 \
   /u01/app/oracle/acfsdata/volume1
```

ORACLE

The COMPATIBLE.ADVM disk group attribute must be 11.2.0 or later before you can create an ADVM volume in the disk group. You can set the attribute with the asmcmd command:

```
setattr -G DATA compatible.advm 11.2.0.0.0
```

To create and mount an ACFS volume, use the procedure shown in the slide.

As a user with the SYSDBA privilege in the ASM instance, perform the following:

1. Create the volume by using the volcreate command.
2. View the volume information to determine the volume name.
3. Make a mount point directory. This is where the volume is mounted. It should have appropriate ownership and permissions.

As the root user:

1. Make the file system. This command formats the volume for an ACFS file system.
2. Mount the volume to the mount point

You may register an ACFS volume. Registering the volume enables the cluster-ready services daemon to mount the volume automatically at startup. The –a option indicates that this volume must be added to the registry. The –f option is used with the add option to allow replacement of an existing registry entry. You may register the volume before or after you mount the volume.

Note: The mkfs command can be used to create other file systems on an ADVM volume. Only one file system is allowed per volume.

Managing Dynamic Volumes with SQL*Plus

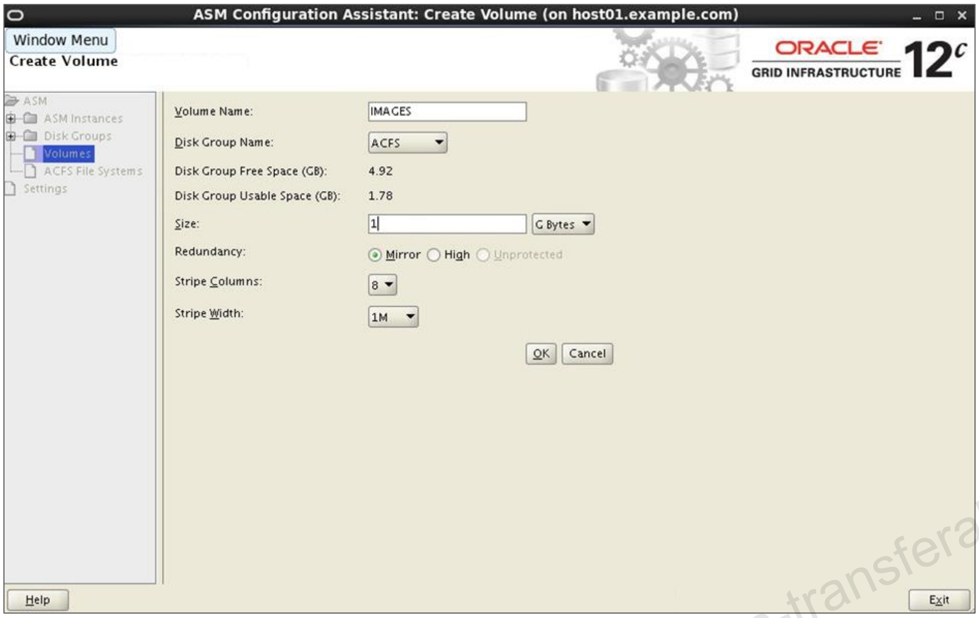
- Command examples:

```
SQL> ALTER DISKGROUP DGROUPA
         ADD VOLUME asmvol1 SIZE 10g;
SQL> ALTER DISKGROUP DGROUPA
         RESIZE VOLUME asmvol1 SIZE 15G;
SQL> ALTER DISKGROUP DGROUPA
         DROP VOLUME asmvol1;
SQL> ALTER DISKGROUP DGROUPA
         ENABLE VOLUME asmvol1;
SQL> ALTER DISKGROUP ALL
         DISABLE VOLUME ALL;
SQL> ALTER DISKGROUP DGROUPA
         MODIFY VOLUME asmvol1 USAGE 'acfs';
```

ORACLE

SQL*Plus has added clauses to the `ALTER DISKGROUP` command that allow you to manage ASM volumes. Notice that each volume is managed as part of a disk group. SQL*Plus allows you to create volumes with the `add` command, `resize` the volumes, and `drop`, `enable`, and `disable` the volumes. The `modify volume` command allows you to set the mount point and the usage name of the volume.

Creating an ACFS Volume with ASMCA

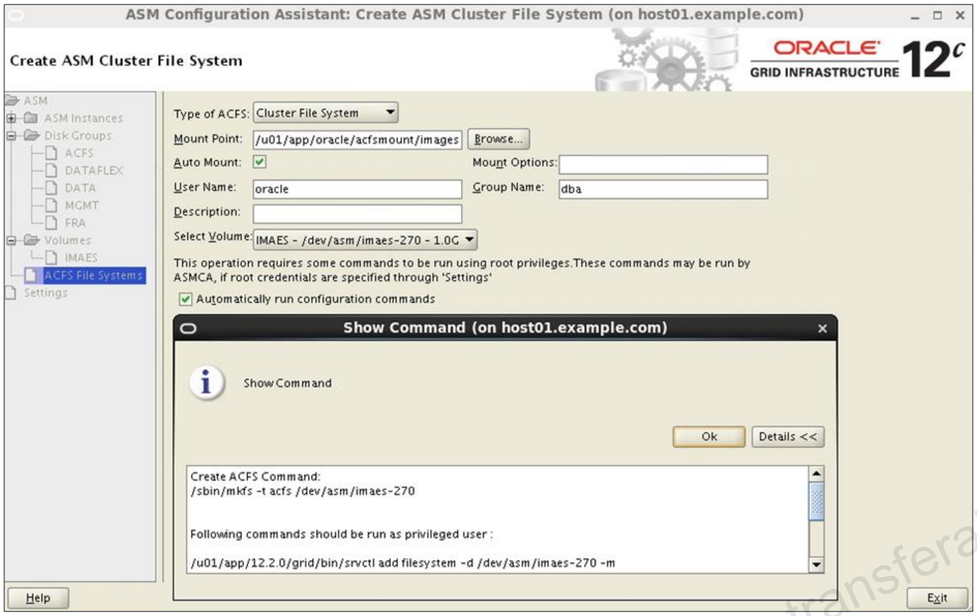


ORACLE

In ASMCA, click the Volume menu in the navigation pane and then, on the Volumes page, click Create. On the Create Volume page, you specify the name, disk group, and size of the volume. Optionally, you can specify the mirroring and striping attributes.

Click OK to create the ACFS volume.

Creating the ACFS File System with ASMCA



After the volume is created, on the Create ASM Cluster File System page, select the volume, the type of file system, and the mount point. ASMCA creates the file system on the volume and registers the volume, but the mount must be performed by the `root` or `administrator` user.

Using ASMCMD for Dynamic Volumes

```
volcreate -G diskgroup -s size
[--redundancy {high|mirror|unprotected}]
[--primary {hot|cold}] [--secondary {hot|cold}] volume

volresize -G diskgroup -s size [ -f ] volume

voldelete -G diskgroup volume

volenable { -a | -G diskgroup -a | -G diskgroup volume }

voldisable { -a | -G diskgroup -a | -G diskgroup volume }

volset -G diskgroup [ --usagestring string]
[--mountpath mount_path ]
[--primary {hot|cold}] [--secondary {hot|cold}] volume

volinfo { -a | -G diskgroup -a | -G diskgroup volume }
volinfo {--show_diskgroup |--show_volume} volumedevice

volstat [-G diskgroup] [volume]
```

ORACLE

The commands shown in the slide allow the ASM volumes to be managed from ASMCMD. The ASMCMD commands allow you to have the same level of control through ASMCMD as you do through the SQL*Plus commands or Enterprise Manager.

Linux-UNIX Extensions

- Create an ACFS file system:
  mkfs [-vf] -t acfs [-b blksz] [-n name] device [blocks]

```
# mkfs -t acfs /dev/asm/vol1-nnn
```

- Mount an ACFS file system:
  mount [-v] -t acfs [-o options] device dir

```
# mount -t acfs /dev/asm/vol1-nnn /oracle/cluster1/myacfs
```

- Unmount an ACFS file system:
  umount [-v] device|dir

```
# umount /oracle/cluster1/myacfs
```

- Check and repair an ACFS file system:
  fsck [-avnf] -t acfs [info] device

```
# fsck -t acfs /dev/asm/vol1-nnn
```


The commands shown in the slide have been extended with additional options to support Oracle ACFS. All other Linux file system commands operate without change for Oracle ACFS.

For example, Oracle ACFS adds a set of Oracle ACFS–specific mount options to those provided with the base operating system platform. You should review both mount options for the Linux platforms in addition to the Oracle ACFS–specific options for the complete set of file system mount options.

File systems on Oracle ADVM volumes that are not Oracle ACFS file systems, such as ext3, are managed with the same Linux commands by using file-specific options for the type of file system.

ACFS Utilities for Multiple Environments

- ACFS commands for multiple environments:

Command	Function
`acfsbg`	Debugs an Oracle ACFS file system
`acfsutil info`	Displays new ACFS file and file system features and information
`acfs plugin disable|enable|info`	Manages the ACFS plug-in infrastructure
`acfsutil snap`	Creates and deletes ACFS snapshots
`acfsutil registry`	Registers an ACFS file system with the ACFS mount registry
`acfsutil rmfs`	Removes an ACFS file system
`acfsutil size`	Resizes an ACFS file system
`acfsutil snap convert|create|delete|info`	Manages snapshots
`acfsutil tune`	Views or modifies ACFS tune-ables
`advmutil canonical`	Displays the canonical name of an Oracle ADVM volume
`advmutil tune`	Modifies or displays Oracle ADVM parameters
`advmutil volinfo`	Displays information about Oracle ADVM volumes

ORACLE

The slide shows a set of commands that are the same on both Windows and Linux/UNIX platforms. The commands listed allow you to display and manipulate ACFS file systems. You can use `acfsutil help` on all platforms to display help text. You can run `acfsutil version` on all platforms to display the Oracle ACFS version.

When the options are entered with commands on a Windows platform, use `/` instead of – with the option. For example, you can display help for `acfsutil` on a Linux platform with `acfsutil -h`. On a Windows platform, use `acfsutil /h`.

A mount point on a Windows operating system can be a drive letter or a directory, including the drive letter. When using a drive letter in a command, include the backslash (`\`) with the drive letter, such as in `M:\`, to avoid the possibility of triggering a Windows path substitution to the last accessed path on the specified drive.

For more information about running Oracle ACFS `acfsutil` commands, refer to *About Oracle ACFS Command-Line Tools in Oracle Automatic Storage Management Administrator's Guide, 12c Release 2 (12.2)*

Configuration Settings for Database Files

- Set the ASM and ADVM compatibility attributes to `12.1`.
  - Required to enable new ASM and ADVM features

```
SQL> [ CREATE | ALTER ] DISKGROUP ...
     ATTRIBUTE 'compatible.asm' = '12.1',
               'compatible.advm' = '12.1';
```

- Set stripe columns to `1`, or set stripe columns to `8` and the stripe width to `1` MB for pre-12.1.0.2 ADVM volumes.

```
ASMCMD> volcreate –G diskgroup –s size --column 1 volume
```

- Volumes that are created while running 12.1.0.2 or later already default to the high performance configuration.
- Set `FILESYSTEMIO_OPTIONS=SETALL` in the database initialization parameter file.
  - Enables direct I/O for the database, bypassing the OS file system cache

ORACLE

When storing database data files on Oracle ACFS, you must set the `FILESYSTEMIO_OPTIONS` initialization parameter to `setall`; other settings are not supported. To achieve optimal performance with database data files, set ASM and ADVM compatibility attributes to 12.1 or higher for the disk group that contains the Oracle ADVM volume that is intended to hold the data files. For volumes created before 12.1.0.2, set the stripe columns to `1`, or set the stripe columns to `8` and the stripe width to 1 MB. Volumes that are created while running 12.1.0.2 or later already default to the high performance configuration (stripe columns = 8 and stripe width = 1 MB).

Support for database data files on Oracle Exadata (Linux) begins with Oracle Grid 12*c* Release 1 (12.1.0.2). However, Oracle ACFS does not currently have the ability to push database operations directly into storage.

Starting with release 12.1.0.2, database files and redo logs can be stored in Oracle ACFS file systems on Windows. `COMPATIBLE.ADVM` must be set to `12.1.0.2`.

ACFS Support for Database File Mapping Views

- Oracle ACFS supports Oracle Database file mapping views to the Oracle ASM device level.
- The `V$MAP` views are only refreshed by executing the `DBMS_STORAGE_MAP.MAP_ALL` procedure.
- Before running any queries on the `V$MAP` views:
  - Ensure that the `FILE_MAPPING` initialization is set to `TRUE`
  - Execute the `DBMS_STORAGE_MAP.MAP_ALL` procedure to build the database I/O mapping information

```
SQL> ALTER SYSTEM SET file_mapping=true;
SQL> EXEC DBMS_STORAGE_MAP.MAP_ALL(10000);
```


With the introduction of Oracle Database 12.1.0.2, ACFS supports Oracle Database file mapping views to the Oracle ASM device level. The following database mapping views are supported by Oracle ACFS:

- `V$MAP_FILE`
- `V$MAP_FILE_EXTENT`
- `V$MAP_ELEMENT`
- `V$MAP_FILE_IO_STACK`

The `V$MAP` views are only refreshed by executing the `DBMS_STORAGE_MAP.MAP_ALL` procedure. The Oracle ACFS file mapping interface does not utilize the external `fmputl` process or its supporting libraries. Oracle ACFS does not provide support for the `V$MAP_SUBELEMENT` view.

Before running any queries on the `V$MAP` views, ensure that the `FILE_MAPPING` initialization is set to `TRUE`, and then run the `DBMS_STORAGE_MAP.MAP_ALL` procedure to build the mapping information for the entire I/O subsystem that is associated with the database. For example, connect as `SYSDBA` to the database instance and run the following:

```
SQL> ALTER SYSTEM SET file_mapping=true;
SQL> EXEC DBMS_STORAGE_MAP.MAP_ALL(10000);
```

Note: Oracle ACFS does not provide support for Oracle Database file mapping on Windows.

To view Oracle ASM information with V$MAP_ELEMENT:

```
SQL> SELECT ELEM_NAME, ELEM_IDX, ELEM_TYPE, ELEM_SIZE, ELEM_DESCR
FROM V$MAP_ELEMENT;

ELEM_NAME ELEM_IDX ELEM_TYPE ELEM_SIZE ELEM_DESCR
------------ ---------- ---------- ---------- --------------------
+/dev/xvdd1 0 ASMDISK 117184512 TEST_0001
+/dev/xvdc1 1 ASMDISK 117184512 TEST_0000
```

Query V$MAP_FILE to view Oracle ACFS data file information:

```
SQL> SELECT FILE_NAME, FILE_MAP_IDX, FILE_TYPE, FILE_STRUCTURE, FILE_SIZE,
FILE_NEXTS FROM V$MAP_FILE WHERE REGEXP_LIKE(FILE_NAME, '*users01.dbf');

FILE_NAME              FILE_MAP_IDX FILE_TYPE FILE_STRU FILE_SIZE FILE_NEXTS
---------------------- ------------ --------- --------- ---------- ---------
/data/orcl/users01.dbf            4 DATAFILE  FILE      10256      41
```

Query V$MAP_FILE_EXTENT to view element and file offset information:

```
SQL> SELECT FILE_MAP_IDX, EXT_NUM, EXT_ELEM_OFF, EXT_SIZE, EXT_FILE_OFF, EXT_TYPE,
ELEM_IDX FROM V$MAP_FILE_EXTENT WHERE FILE_MAP_IDX=4;

FILE_MAP_IDX EXT_NUM EXT_ELEM_OFF EXT_SIZE   EXT_FILE_OFF EXT_TY   ELEM_IDX
------------ ------- ------------ ---------- ------------------- ----------
           4       0     58105664        192            0 DATA            0
           4       1     58154752        256          192 DATA            1
           4       2     58089472        256          448 DATA            0
...
           4      39     58140928        256         9920 DATA            1
           4      40     58108160         88        10176 DATA            0

41 rows selected.
```

ACFS Cluster Resources

- File system resource enhancements

```
$ srvctl add filesystem -device vol_device –mountpointpath mount_path
  [-volume vol_name] [-diskgroup dg_name] [-user user]
  [-nodes node_list | -serverpools serverpool_list]
  [-fstype {ACFS|EXT3|EXT4}] [-fsoptions options]
  [-description description] [-appid application_id]
  [-autostart {ALWAYS|NEVER|RESTORE}]
```

- New ADVM resource
  – Completes the storage resource dependency tree
  – Ensures resources are started and stopped in the right order
- ACFS mount registry resource removed
  – All file system attributes are in the file system resource.
- Consistent file system classification
  – There is no difference between general file systems and the file systems that contain Oracle Database home directories.

ORACLE

Oracle Database 12*c* provides the following enhancements to cluster resources for ACFS:

- **File System Resource Enhancements**

  The file system resource has been enhanced through the provision of extra attributes. The file system resource enhancements provide administrators with better control over where file systems are mounted, what mount options are used, and whether the file system should be mounted automatically. The full form of the `srvctl add filesystem` command is shown in the slide. The new attributes include:

`-nodes`	Comma-separated list of nodes on which the file system will be mounted (The default is all nodes.)
`-serverpools`	Comma-separated list of server pools on which the file system will be mounted (The default is all server pools.)
`-fstype`	File system type
`-fsoptions`	Comma-separated list of file system mount options
`-description`	File system description
`-appid`	File system application ID
`-autostart`	Policy for automatically starting the file system

- **New ADVM Resource**

  A new resource type is included for ADVM resources. The new ADVM resource completes the storage resource dependency tree that includes ASM, ADVM, and ACFS. It allows more precise control over resource dependencies to ensure that resources are managed correctly, including the correct order for resource startup and shutdown. The ADVM resource is created automatically when a volume is created, and it contains no adjustable attribute settings. The current status of volume resources can be determined by using the `srvctl status volume` or `crsctl status resource` commands.

- **ACFS Mount Registry Resource Removed**

  In earlier releases, a Cluster Ready Services (CRS) resource was associated with the ACFS mount registry. This resource was primarily used to ensure that file systems were automatically mounted after a system restart. In addition, CRS resources were also associated with ACFS file systems that were designated as Oracle Database home file systems.

  Using the file system resource enhancements provided in Oracle Database 12*c*, all file system attributes previously maintained in the ACFS mount registry can be specified in the ACFS file system resource, and the ACFS mount registry resource is no longer required.

  All the ACFS registry interfaces and functions are preserved in Oracle Database 12*c*; however, the file system attributes are stored in the ACFS file system resource that is associated with each file system.

- **Consistent File System Classification**

  In earlier releases, an ACFS file system could be configured as a general file system or as an Oracle Database home file system. With Oracle Database 12*c*, there is no difference between general file systems and the file systems that contain Oracle Database home directories. That is, any ACFS file system can house Oracle Database home directories and other data files, and no additional configuration is required to enable storage of Oracle Database home directories.

Implementing Node-Specific File System Dependencies

- Use Case: A clustered application needs to record log file information separately for each node.
- Implementation example:

```
$ srvctl add filesystem -device /dev/asm/log1-123
  -mountpointpath /mnt/logn01 -appid LOGFS -node c00n01
$ srvctl add filesystem -device /dev/asm/log2-123
  -mountpointpath /mnt/logn02 -appid LOGFS -node c00n02

$ crsctl status type | grep LOGFS
TYPE_NAME=ora.LOGFS_fs.type

$ crsctl modify resource my_application
  -attr "START_DEPENDENCIES=hard(type:ora.LOGFS_fs.type)
  pullup(type:ora.LOGFS_fs.type)"
```


The new `appid` file system resource attribute can be used to define dependencies between a clustered application and the separate node-specific file systems that are running on each cluster node. A common use case occurs when a clustered application needs to record log file information separately for the application instances that are running on each node.

The slide shows an implementation example based on a two-node cluster (`c00n01` and `c00n02`). The example assumes that two volumes (`/dev/asm/log1-123` and `/dev/asm/log2-123`) have already been formatted with ACFS.

The `srvctl add filesystem` commands create separate node-specific file system resources. Note that –`appid LOGFS` is specified in both commands.

Setting the `appid` file system resource attribute results in the creation of a type containing `appid` in the type name. The `crsctl status type` command can be used to identify the complete type name, which is `ora.LOGFS_fs.type` in this example.

Finally, the type name can be used in a dependency definition that is associated with an application that requires the file systems. In this example, a cluster resource named `my_application` is modified to depend on the file systems associated with the `ora.LOGFS_fs.type` type.

As a result of this configuration, when `my_application` starts on either cluster node, the corresponding file system is also mounted (`/dev/asm/log1-123` on `c00n01` and `/dev/asm/log2-123` on `c00n02`).

ACFS Snapshots

ACFS snapshots:

- Are space efficient. Snapshots store:
  - Metadata
  - Original versions of changed blocks
  - Using sparse files
- Can be used to:
  - Revert to a version in the snapshots
  - Recover a deleted file
  - Back up a consistent data set
- Are limited to 63 snapshot views per volume
- Require the disk group compatibility attribute for ADVM set to 11.2.0.3.0 or later

An Oracle ACFS snapshot is an online, read-only or read-write, point in time copy of an Oracle ACFS file system. The snapshot copy is space-efficient and uses Copy-On-Write functionality. Before an Oracle ACFS file extent is modified or deleted, its current value is copied to the snapshot to maintain the point-in-time view of the file system.

Oracle ACFS snapshots are immediately available for use after they are created. The snapshots are created in the `.ACFS/snaps/` directory of the file system. They are always online while the file system is mounted. Consequently, an Oracle ACFS snapshot can support online recovery of files that are inadvertently modified or deleted from a file system. With up to a total of 63 read-only, read-write, or a combination of read-only and read-write snapshot views supported for each file system, flexible online file recovery solutions that span multiple views can be employed. An Oracle ACFS snapshot can also be used as the source of a file system backup, because it can be created on demand to deliver a current, consistent, online view of an active file system.

To use Oracle ACFS read-write snapshots, the disk group compatibility attribute for ADVM must be set to 11.2.0.3.0 or later. If you create a read-write snapshot on an existing Oracle ACFS file system from a version earlier than 11.2.0.3.0, the file system is updated to the 11.2.0.3.0 or later format. After a file system has been updated to a later version, an Oracle ACFS file system cannot be reverted to an earlier version, and accordingly cannot be mounted on an earlier Oracle Grid Infrastructure version.

Managing ACFS Snapshots

Manage ACFS snapshots with `acfsutil` commands.

- Create snapshots:

```
$ acfsutil snap create snapshot_2 /u01/app/oracle/acfsdata/testvol
```

- Delete snapshots:

```
$ acfsutil snap delete snapshot_2 /u01/app/oracle/acfsdata/testvol
```

- View file system information, including snapshots:

```
$ acfsutil info fs mount_point ls -l mount_point/.ACFS/snaps
```

ORACLE

ACFS snapshots can be managed with the `acfsutil snap` commands to create and delete snapshots. You can view the contents of the snapshots by using standard OS commands to search and view the contents of a directory.

An example of creating a snapshot on the ACFS volume that is mounted at `/u01/app/oracle/acfsdata/testvol` with a snapshot name of snapshot_2 is:

```
acfsutil snap create snapshot_2 /u01/app/oracle/acfsdata/testvol
```

This `create` command creates a directory named:

```
/u01/app/oracle/acfsdata/testvol/.ACFS/snaps/snapshot_2
```

The snapshot is immediately available for use.

You can view the file system information, including the number of snapshots with:

```
acfsutil info fs /u01/app/oracle/acfsdata/testvol
```

```
/u01/app/oracle/acfsdata/images
    ACFS Version: 12.2.0.1.0
    flags:        MountPoint,Available
    mount time:   Tue Aug 13 17:07:32 2013
    volumes:      1
    total size:   1073741824
    total free:   951771136
    primary volume: /dev/asm/testvol-33
        label:
        flags:                 Primary,Available,ADVM
        on-disk version:       39.0
        allocation unit:       4096
        major, minor:          251, 16897
        size:                  1073741824
        free:                  951771136
        ADVM diskgroup         DATA
        ADVM resize increment: 33554432
        ADVM redundancy:       mirror
        ADVM stripe columns:   4
        ADVM stripe width:     131072
    number of snapshots:  1
    snapshot space usage: 32768
    replication status: DISABLED
```

You can see the names of the snapshots with the OS command:

```
ls /u01/app/oracle/acfsdata/testvol/.ACFS/snaps
```

Snapshots of Snapshots

- Existing snapshots can be used as the source for a new snapshot.
- Any combination of read-only and read-write snapshots
  - An ACFS file system can have up to 1023 read-only, read-write, or combination of read-only and read-write snapshot views per file system.
  - Command syntax:

```
$ acfsutil snap create [-w|-r] –p parent_snap_name snap_name
  mountpoint
```

- Conversion between read-only and read-write snapshots
  - Command syntax:

```
$ acfsutil snap convert -w|-r snap_name mountpoint
```

ORACLE

With Oracle Database 12*c*, ACFS supports the creation of snapshots based on an existing snapshot of the same ACFS file system, which is otherwise known as Snaps-of-Snaps. Any combination of read-only and read-write snapshots is supported. For example, a read-write snapshot can be based on an existing read-only snapshot, and a read-only snapshot can be based on an existing read-write snapshot.

Oracle ACFS supports 1023 snapshots on 64-bit systems. These snapshots can be any combination of read-only and read-write snapshots. To support the creation of 1023 snapshots, the disk group compatibility attribute for ADVM must be set to 12.1.0.2 or later. Before or after setting `COMPATIBLE.ADVM` to 12.1.0.2, all existing snapshots must be deleted or the limit remains at 63 snapshots.

Also, snapshot conversions are enabled between read-only and read-write snapshots. Conversion in either direction is supported. For example, a read-only snapshot can be converted to a read-write snapshot, then modified, and finally converted back to a read-only snapshot.

ACFS Backups

- An ACFS file system may be backed up by using:
  - Standard OS file system backup tools
  - Oracle Secure Backup
  - Third-party backup tools
- ACFS snapshots present a stable point-in-time view.
- Backup applications that use interfaces other than the standard read/write OS interfaces are not supported.

Oracle ACFS runs on operating system platforms as a native file system technology that supports native operating system file system APIs. Consequently, backup applications that access files by using the native OS file system interfaces are able to access and back up Oracle ACFS file systems and other native operating system file systems. Oracle ACFS snapshots can be dynamically created and used to present a consistent, online view of an active file system to a backup application.

Backup applications that use interfaces other than the standard operating system interfaces (read or write) are not supported with Oracle ACFS. For example, Windows backup applications that depend on the presence of reparse points or the Windows Volume Shadow Copy Service (VSS) are not supported.

ACFS Performance

ACFS performance benefits from:

- Using larger `write()` sizes, such as 8 K or larger
- Distribution and load balancing of ASM file segments
- ACFS file extents distributed across ASM file segments
- User and metadata caching
- In-memory updates of transaction logs
- Using Deadline I/O Scheduler for the disks in the disk group on a Linux system

The ACFS storage allocation and I/O access schemes benefit from the distribution and load balancing of ASM file segments across the ASM disks in a disk group. ACFS storage allocation further distributes individual ACFS file extents across ASM file segments for increased distribution of individual file data and file I/O parallelism. ACFS user and metadata caching techniques accommodate a variety of non-database application workload I/O access patterns with high cache hit rates that result in reduced I/O traffic to the underlying ASM file and deferred updates of modified file cache data. The ACFS transaction logs are first updated in memory, deferring the actual transfer of log and file data through periodic updates to the associated ASM file.

ACFS Views

View	Description
V$ASM_ACFS_ENCRYPTION_INFO	Contains encryption information for each ACFS file system
V$ASM_ACFS_SECURITY_INFO	Shows information about every realm in the ACFS security realm for each ACFS file system
V$ASM_ACFS_SEC_CMDRULE	Shows information about ACFS security command rules
V$ASM_ACFS_SEC_REALM	Shows security realm information for each ACFS file system
V$ASM_ACFS_SEC_REALM_GROUP	Shows group information in the ACFS security realm
V$ASM_ACFS_SEC_REALM_USER	Shows user information in the ACFS security realm
V$ASM_ACFS_SEC_RULE	Shows information for every ACFS security rule
V$ASM_ACFS_SEC_RULESET	Shows information for every ACFS security ruleset
V$ASM_ACFSSNAPSHOTS	Shows snapshot data for each mounted ACFS file system
V$ASM_ACFSTAG	Contains tag data about files on mounted ACFS file systems
V$ASM_ACFSVOLUMES	Contains information about mounted Oracle ADVM volumes
V$ASM_FILESYSTEM	Contains information about every mounted ACFS file system
V$ASM_VOLUME	Contains information about ADVM volumes that are members of an ASM instance
V$ASM_VOLUME_STAT	Contains statistical information for each ADVM volume

The table in the slide contains some of the views that can be used to obtain information about Oracle Automatic Storage Management Cluster File Systems. These views are accessible from the Oracle ASM instance. To display information about ACFS file systems or volumes that are located on nodes in an Flex ASM configuration, you must connect to the ASM proxy instance instead of the local Oracle ASM instance.

Note: When viewing space usage values in ACFS views on Windows systems, the values may differ from the sizes in Windows folders. The mechanism used by Folder Properties on Windows systems only accounts for files and should be considered an approximate value.

Summary

In this lesson, you should have learned how to:

- Administer ASM dynamic volume manager
- Manage ASM volumes
- Implement ASM cluster file system (ACFS)
- Manage ACFS with various tools
- Use ACFS snapshots

Practice 8 Overview: Administering Oracle ACFS

This practice covers the following topics:

- Managing an ACFS file system
  - Create
  - Register
  - Mount
- Managing ACFS Snapshots

ORACLE

9

ACFS/ADVM Enhancements

Objectives

After completing this lesson, you should be able to:

- Understand the following ACFS/ADVM enhancements in 12.2
  - Oracle ACFS Automatic Resize
  - Oracle ACFS Scrubbing
  - Oracle ACFS Defragger
  - Oracle ACFS Metadata Collection
  - Oracle Metadata Acceleration
  - 4K Sector Support for Oracle ACFS and Metadata
  - Oracle ACFS Compression
  - Oracle ACFS Spare Files
  - Oracle ACFS Loopback Devices

ORACLE

Oracle ACFS Automatic Resize

- The `acfsutil size` command has been enhanced to support auto resizing of an ACFS file system.
- Auto resizing allows you to specify an increment by which an ACFS file system will grow if available free space falls below a specified amount.

```
$ acfsutil size -h
Usage: acfsutil [-h] size [[-|+]nnn[K|M|G|T|P]] [-a nnn[K|M|G|T|P]] [-x
     nnn[K|M|G|T|P]] [-d <device>]  <path>
                                 - Resize file system and configure
                                   auto-resize
                [-a]             - Auto-resize increment
                [-x]             - Auto-resize maximum
                [-d]             - Resize only this device

$ acfsutil size -a 1G -x 10g /acfsmounts/acfs1
acfsutil size: ACFS-03642: successfully updated auto-resize settings
```

- Automatic resizing requires the ADVM compatibility attribute to be set to 12.2 or later.

ORACLE

The `acfsutil size` command has been enhanced to support auto resizing of an ACFS file system. Auto resizing allows a user to specify an increment by which an ACFS file system will automatically grow if the amount of available free space in the file system falls below a specified amount.

There is also an option to specify the maximum size allowed when using the automatic resize option. The output of the `acfsutil info fs` command displays the automatic resize increment and maximum amounts.

The automatic resize option, `-a`, specifies an increment by which an Oracle ACFS file system grows automatically if the amount of available free space in the file system falls below the amount specified by the increment. You can also specify the maximum size (`-x` option) allowed when using the automatic resize option. The output of the `acfsutil info fs` command displays the automatic resize increment and maximum amounts. If only the `-x` option is provided without the `-a` option (and no automatic resize increment has previously been specified), a warning is issued if the increment is still zero. If an increment has previously been set by an earlier invocation of the command, the command succeeds. Automatic resizing requires the ADVM compatibility attribute to be set to 12.2 or later.

Oracle ACFS Automatic Resize

- If an automatic resize fails on a node for any reason, automatic resizing on the node is temporarily disabled.
  - Automatic resizing may still take place on another node.
  - If the failure is due to the underlying volume being out of free space, all the nodes quickly disable automatic resize.
  - The file system continues to periodically try to resize at increasingly longer intervals.
  - After a resize succeeds, (when space is added to the disk group, for example), automatic resize is re-enabled on the node.
- Automatic resize can be manually re-enabled by performing either of the following:
  - Re-mounting the file system
  - Running the `acfsutil size` command again with the `-a` or `-x` option to update the automatic resize settings

If an automatic resize fails on a node for any reason, automatic resizing on the node is temporarily disabled. Automatic resizing may still take place on another node. However, if the failure is due to the underlying volume being out of free space, all the nodes quickly disable automatic resize. The file system continues to periodically try to resize at increasingly longer intervals. After a resize succeeds, such as when more space is added to the underlying disk group, automatic resize is re-enabled on the node. Automatic resize can be manually re-enabled by either re-mounting the file system or running the `acfsutil size` command again with the `-a` or `-x` option to update the automatic resize settings. When a file system has grown to the maximum size specified with `acfsutil size -x`, auto resize remains enabled but does not grow the file system further until the maximum is increased.

The `-d` argument is not compatible with the `-a` and `-x` automatic resize arguments. Unless a volume device is specified with `-d`, it is assumed that the caller is specifying an amount of storage that applies to the primary volume, which represents the size of the file system. When growing a file system with an accelerator volume, the `acfsutil size` command determines if the accelerator size needs to be increased. If the accelerator must be larger and cannot be increased, the command fails and the primary volume remains at the same size.

Oracle ACFS Scrubbing

- The `acfsutil scrub` command checks for and reports any inconsistencies in the metadata or file data.
- For paths with metadata inconsistencies, the scrubber displays the inconsistent path.
- For file paths with user data inconsistencies, the scrubber provides the following information:
  - Path name
  - Byte offset and length of inconsistency
  - ADVM volume byte offset of inconsistency
- ACFS `acfsutil scrub` examples include:

```
$ /sbin/acfsutil scrub -m /acfsmnt/textdoc1.txt
$ /sbin/acfsutil scrub -b txt,pdf -z /acfsmnt/dir1
$ /sbin/acfsutil scrub -p 5 -t 3 /acfsmnt/dir1
```


Oracle ACFS provides the scrubbing functionality with the `acfsutil scrub` command, which checks for and reports any inconsistencies in the metadata or file data. On Linux and AIX systems, the `acfsutil scrub` command checks the mirror consistency of the Oracle ACFS directory metadata and file user data and metadata. You must have `root` privileges to run `acfsutil scrub`. For paths with metadata inconsistencies, the scrubber displays the inconsistent path. For file paths with user data inconsistencies, the scrubber provides the following information:

- Path name
- Byte offset of inconsistency
- Byte length of inconsistency
- ADVM volume byte offset of inconsistency

The examples in the slide illustrate `acfsutil scrub` usage. The first command scrubs only the metadata of the specified file. The second command scrubs the specified directory and all nested files, except snapshots, `.txt` files, and `.pdf` files. The third command scrubs the specified directory and all nested files with the power level set to 5 and trace level set to 3.

Oracle ACFS Fragmentation

- Databases that share storage with snapshots can become fragmented under active OLTP workloads.
- This can cause the location of data in the volume to be discontiguous for sequential scans.
- Defragmentation may also be needed when database data files reside in file systems enabled with ACFS compression.
- ACFS automatically defragments these files in the background.
- Fragmentation is reported through:
  - The `acfsutil defrag dir|file` commands
  - The `acfsutil info file|fs` commands

Databases that share storage with snapshots or with the base of the file system can become fragmented under active online transaction processing (OLTP) workloads. This fragmentation can cause the location of data in the volume to be discontiguous for sequential scans. ACFS automatically defragments these files in the background. Fragmentation is reported through the `acfsutil defrag dir` and `acfsutil defrag file` commands, and it can also be viewed with the `acfsutil info file` and `fs` commands.

```
$ acfsutil info fs -f
/mnt/oracle/rhpimages/chkbase
Free space allocation:
   Size       Count   Percent    Total Size
    1G:          1     91.18   15462940672
    128M:        2      4.78     810110976
    5M:         20      3.49     591540224
    1M:         24      0.41      70139904
    256K:       27      0.08      14041088
    128K:       16      0.02       3174400
    64K:        21      0.01       2289664
    32K:        26      0.01       1306624
    16K:        37      0.00        827392
    12K:         5      0.00         61440
    8K:         54      0.00        442368
    4K:        231      0.01        946176
File system size:      26843545600   (  25.00 GB )
Total free:            16957820928   (  15.79 GB )
Percentage of free space less than 128K: 0.03%
```

The `acfsutil defrag dir` and `file` commands also enable on-demand defragmentation of a file in the event that automatic defragmentation does not occur quickly enough. Defragmentation may also be needed when database data files reside in file systems that are enabled with ACFS compression.

Oracle ACFS Defragger

- Live Oracle ACFS file systems can be defragged on demand by using the `acfsutil defrag dir |file` commands.
  - The `acfsutil defrag dir` command is used to defragment files in a specified subdirectory.

```
# acfsutil defrag dir -h
Usage: acfsutil [-h] defrag dir [-r] <path>
        Defrag all files in a specific directory
        [-r] recursively traverse directories
        <path> path to a specific dir

$ /sbin/acfsutil defrag dir /acfsmounts/critical_apps/01jan1016
```

  - The `acfsutil defrag file` command defragments files.

```
# acfsutil defrag file -h
Usage: acfsutil [-h] defrag file [-i] <path> [<path> ...]
                     Defrag specific files
                     [-i] Estimate reduction in number of extents
    possible
                     <path> path to a specific file

$ /sbin/acfsutil defrag file /acfsmounts/critical_apps/testfile
```


Oracle ACFS provides the on-demand defragging functionality that can be run on a live Oracle ACFS file system by using the `acfsutil defrag dir` and `acfsutil defrag file` commands. In addition, the automatic database file defragger has been updated for scalability with support for concurrent, multiple-node, database file defragmentation.

The `acfsutil defrag dir` command is used to defragment files in a specified subdirectory. The `–r` option is used to recursively defragment files in the subdirectories of a specified directory.

The `acfsutil defrag file` command defragments files. The `–i` option displays the estimated reduction in the number of extents for the files to be defragmented.

Administrator privileges are required to use these commands or you must be a member of the Oracle ASM administrator group and must have write access to the files to be defragmented.

Oracle ACFS Metadata Collection

- The ACFS metadata collector copies metadata into a separate output file that can be used for analysis and diagnostics.
  - The `acfsutil meta` command is used to collect the metadata of an Oracle ACFS volume.

```
# acfsutil meta -f /u01/mymetafile /dev/asm/testvol-180
 Reading configuration information from device /dev/asm/testvol-180 ...
  Log file name: acfs.meta.log
  Input device name: /dev/asm/testvol-180
  Output file name: /u01/mymetafile

  Copying first 10 megabytes from device /dev/asm/testvol-180 ...
  Continue scanning metadata on device /dev/asm/testvol-180 ...
  End of general metadata scan on device /dev/asm/testvol-180 ...

  Copying global bit map from device /dev/asm/testvol-180 ...
  Copying snap map bit map from device /dev/asm/testvol-180 ...
  Copying local bit map from device /dev/asm/testvol-180 ...

acfsutil meta summary:
  Log file name: acfs.meta.log
  Input device name: /dev/asm/testvol-180
  Output file name: /u01/mymetafile
```

ORACLE

The Oracle ACFS metadata collector tool copies selected Oracle ACFS metadata structures into a new, separate output file that can be used for analysis and diagnostics. The `acfsutil meta` command is provided for collecting the metadata of an Oracle ACFS volume.

The `acfsutil meta` command partially copies an Oracle ACFS file system into a separate specified output file. The metadata collector reads the contents of the file system specified by the volume device name of an Oracle ACFS file system. The input file system is searched for Oracle ACFS metadata, and then all metadata that is found is written into the specified output file. The generated output file can be used for further diagnostics and analysis, without impact to the original file system at the customer site.

If the original file system is very large, the output file can also be very large. The output file should be placed on a file system that supports sparse files because this placement can reduce the size of the file. When copying the output file, use a utility that supports sparse files. Compress the output file when possible to reduce storage space and transmission time.

If the file system has an accelerator device associated with it, `acfsutil meta` also copies the accelerator device data to a second output file. The second file uses the file name from the `-f` option with an appended `.acc` extension. For example, if you specify `acfsutil meta -f /tmp/mymeta volume1-123`, the metadata collector places a copy of `volume1-123` in `/tmp/mymeta`, and the copy of its accelerator device in the `/tmp/mymeta.acc` file. This operation occurs automatically.

In most circumstances, `acfsutil meta` automatically copies the accelerator device to the second file. However, if you have think that the metadata collector will not be able to find the accelerator device on its own, you can specify the name on the command line with the `-a` option. For example, this situation could occur if the file system is corrupt. Note that using the `-a` option overrides how the metadata collector operates automatically, so `-a` should be used carefully.

The output file should not be placed on the Oracle ACFS device that is specified as the input device because the metadata command might process the output file also. The file should be placed on a file system that can support an output file that is the size of the Oracle ACFS input volume device.

Oracle ACFS Metadata Acceleration

- With the introduction of Oracle Database 12.2, Oracle ACFS supports accelerator metadata storage.
- An accelerator volume can improve performance by reducing the time to access and update Oracle ACFS metadata.
- The accelerator volume is created by using the `-a` option with the `mkfs` command.
  - The value of `COMPATIBLE.ADVM` must be at least 12.2 to use the `-a` option.

```
$ /sbin/mkfs -t acfs /dev/asm/volume1-168 -a /dev/asm/volume2-141
```

- Create the accelerator volume on a disk group with significantly faster storage than the primary disk group storage.
- The size of the accelerator volume should be at least 32 MB for each node in the cluster.

With the introduction of Oracle Database 12.2, Oracle ACFS supports accelerator metadata storage. Use an accelerator volume to improve performance by reducing the time to access and update Oracle ACFS metadata. For example, Oracle ADVM region logging (DRL) files can be optionally stored on accelerator storage, further optimizing I/O performance for the Oracle ADVM storage that hosts the Oracle ACFS file systems. The accelerator volume is created by using the `-a` option with the `mkfs` command. The value of `COMPATIBLE.ADVM` must be at least 12.2 to use the `-a` option.

Create the accelerator volume on a disk group with storage that is significantly faster than the primary volume's storage. For example, Solid State Disk (SSD) storage could be used. Oracle ADVM volumes are created with the ASMCMD `volcreate` command.

The size of the accelerator volume should be at least 32 MB for each node in the cluster. For example, in an eight-node cluster, the recommended minimum size of the accelerator volume is 256 MB. If an accelerator volume is smaller than the recommended size, the file system can operate normally, but the full benefit of the storage accelerator may not be achieved. The minimum size of the accelerator volume is 128 MB.

Accelerator Volume Considerations

- The recommended size of the accelerator volume depends on the workload.
- It is particularly helpful for files with many extents, especially if the extent metadata is updated frequently.
  - You can use the `acfsutil info file` command to view a report on a file's extents.

```
$ /sbin/acfsutil info file file_name
```

- Database files generally have many extents and when ACFS snapshots are in use, the extent metadata is updated often.
- A workload that greatly benefits from an accelerator is a compressed file system.
- `acfsutil size` can be used to automatically grow the accelerator as needed along with the primary volume.

ORACLE

The recommended size of the accelerator volume depends on the workload. It is particularly helpful for files with many extents, especially if the extent metadata is updated frequently. You can use the `acfsutil info file` command to view a report on a file's extents. Database files generally have many extents and when Oracle ACFS snapshots are in use, the extent metadata is updated frequently. A workload that greatly benefits from an accelerator is a compressed file system.

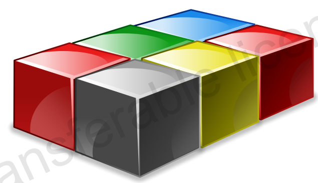
If Oracle ACFS cannot allocate space on the accelerator for critical metadata, the metadata is stored on the primary volume instead. Depending on the frequency with which the metadata is updated, it can have a disproportionate impact on performance. If the slow metadata must be written in the same transaction as the fast metadata, the slow metadata brings down the performance of the entire operation.

The recommended starting size of the accelerator is minimally 0.6% of the size of the file system. If many snapshots are in use that represent several points in time for a database workload, the recommendation is an additional 0.4% per snapshot. For example, a file system with five snapshots may need an accelerator whose size is 2.6% of the size of the primary volume. `acfsutil size` can be configured to automatically grow the accelerator as needed along with the primary volume. The accelerator increases in units of 64 MB. The minimum size of the accelerator volume is 256 MB.

The `mkfs` command requires that the initial accelerator size be at least 0.4% of the size of the primary volume. The accelerator volume is linked to the primary volume that is specified with the `mkfs` command. When mounting a file system, only the primary volume is specified. If the accelerator volume becomes inaccessible for any reason after a file system with the volume is mounted, the file system is taken offline. Only one storage accelerator volume can be associated with an Oracle ACFS file system. After an accelerator volume is associated with a file system, the volume cannot be disassociated from the file system.

4K Sector Support for Oracle ACFS and Metadata

- Oracle ACFS generally formats its metadata block size to be 512 bytes when the logical disk sector is 512 bytes.
- Oracle ACFS formats its metadata block size to be 4096 bytes when the logical disk sector size 4096 bytes.
- Oracle ACFS formats file systems with 4K-sized metadata when the logical disk sector size of the Oracle ADVM volume is 512 bytes and if the `mkfs … -i 4096` option is specified.
- User data IO continues to support transfers as small as 512 bytes for normal user IO requests.



By default, Oracle ACFS formats the file system metadata based on the logical disk sector size of the Oracle ADVM volume. Oracle ACFS generally formats its metadata block size to be 512 bytes when the logical disk sector is 512 bytes. Oracle ACFS formats its metadata block size to be 4096 bytes when the logical disk sector size is 4096 bytes. In addition, Oracle ACFS formats file systems with 4096-byte-sized metadata when the logical disk sector size of the Oracle ADVM volume is 512 bytes and if the `-i 4096` option is specified.

User data IO continues to support transfers as small as 512 bytes for normal user IO requests. When the Oracle ADVM volume of the file system has a logical disk sector size of 4K, user direct IO requests should be aligned on 4K file offsets and be multiples of 4 KB lengths for best performance. Note that only 4K metadata block size formats support ADVM volumes with 4K logical disk sectors.

Both the ACFS primary file system volume and the ACFS accelerator volume are set to the same metadata structure block size. When both primary and accelerator volumes have the same logical disk sector size, `mkfs` creates a file system based on that sector size by default. The `mkfs -i 4096` switch must be specified when one of the volumes has a 4K logical sector size and the other has a 512 byte logical sector size or when both volumes have a 512 byte sector size and you want to create the file system with the 4 KB metadata structures.

The `COMPATIBLE.ADVM` value must be set to 12.2 or later to format an Oracle ADVM volume whose logical disk size is 4096 or to use the `-i 4096` option to format an Oracle ADVM volume whose logical disk sector size is 512 bytes.

4K Sector and Metadata Support Considerations

- COMPATIBLE.ADVM must be set to 12.2 or later to:
  - Format an ADVM volume whose logical disk size is 4096 bytes.
  - Use the -i 4096 option to format an ADVM volume whose logical disk sector size is 512 bytes.
- The -i 4096 option is provided:
  - With the acfsformat command on Windows
  - With the mkfs command in AIX, Linux, and Solaris environments
- The ACFS driver must be loaded for mkfs to execute successfully.

```
# /sbin/mkfs -t acfs /dev/asm/testvol-180 -i 4096
mkfs.acfs: version                   = 12.2.0.0.2
mkfs.acfs: on-disk version           = 46.0
mkfs.acfs: volume                    = /dev/asm/testvol-180
mkfs.acfs: volume size               = 536870912  ( 512.00 GB )
mkfs.acfs: Format complete.
```


Both the Oracle ACFS primary file system volume and the ACFS accelerator volume are set to the same metadata structure block size. When both primary and accelerator volumes have the same logical disk sector size, mkfs creates a file system based on that sector size by default. The mkfs -i 4096 switch must be specified when one of the volumes has a 4K logical sector size and the other has a 512 byte logical sector size or when both volumes have a 512 byte sector size and you want to create the file system with the 4 KB metadata structures.

The COMPATIBLE.ADVM value must be set to 12.2.0.0 or later to format an Oracle ADVM volume whose logical disk size is 4096 or to use the -i 4096 option to format an Oracle ADVM volume whose logical disk sector size is 512 bytes.

The -i 4096 option is provided with the acfsformat command on Windows and the mkfs command in AIX, Linux, and Solaris environments. Use mkfs to create the on-disk structure needed for an Oracle ACFS file system to be mounted. The mkfs command is the traditional Linux command used to build a file system. After mkfs runs successfully, the USAGE column in the V$ASM_VOLUME view displays ACFS. The root privilege is not required; the ownership of the volume device file dictates who can run this command. The Oracle ACFS driver must be loaded for mkfs to operate successfully.

Accelerator Volumes and Automatic Resize

- When using an accelerator volume and automatic resize:
  - The increment specified with the automatic resize `acfsutil size -a` argument applies to the primary volume.
  - However, both the primary volume and the accelerator volume are monitored.
- Automatic resize grows the primary volume by the specified increment, or more if needed.
- Automatic resize monitors and grows the accelerator volume as needed by smaller increments.
- When automatic resize needs to grow the primary volume:
  - It first attempts to grow the accelerator if insufficient space remains on the accelerator to accommodate the new growth
  - If this fails, the primary volume remains at the same size

ORACLE

When using an accelerator volume and automatic resize, the increment specified with the automatic resize `-a` argument applies to the primary volume; however, both the primary and accelerator volumes are monitored. Automatic resize grows the primary volume by the specified increment, or more if needed. Automatic resize monitors and grows the accelerator volume as needed as well, by smaller increments.

When automatic resize needs to grow the primary volume, it first attempts to grow the accelerator if insufficient space remains on the accelerator to accommodate the new growth. If this fails, the primary volume remains at the same size. In addition to being triggered due to free space falling below the configured automatic resize increment, an automatic resize also occurs in other situations, such as the following:

- If the free space in a file system falls below 10% of the current file system size or 5 GB, whichever is smaller
- If a copy-on-write cannot find enough contiguous space in the file system to allocate new storage for the write
- If the defragger cannot find enough contiguous space to allocate in order to defragment 8M worth of extents in a file

All the previously listed conditions protect against the case where a large file system might have enough free space in total that an automatic resize does not appear necessary, but the space is so fragmented that it is not usable by applications.

When a non-sparse file is resized so that it grows by more than the configured auto resize interval, the file system automatically resizes by a multiple of the automatic resize increment so that the resize request can be satisfied.

Oracle ACFS Compression

- ACFS compression can be enabled on ACFS file systems for:
  - Database data files
  - RMAN files
  - Archive logs
  - Data pump files
  - General purpose files

- Redo logs, flashback logs, and control files are *not* compressed.
- ADVM disk group compatibility must be set to 12.2 or later.
- For database-related files, the Oracle Database release must be 11.2.0.4 or later.
- Direct IO, as is done for database files, is done inline and the database file is compressed immediately when written.

ORACLE

Oracle ACFS compression is enabled on a specified Oracle ACFS file system for database data files, RMAN files, archive logs, data pump files, and general purpose files. Redo logs, flashback logs, and control files are not compressed. For database-related files, the Oracle Database release must be 11.2.0.4 or later.

Direct IO, as is done for database files, is done inline and the database file is compressed immediately when written. Cached IO compression is performed asynchronously, after the application has written to the file.

When compression is enabled on a file system, the existing files are not compressed; only newly created files are compressed. When compression is disabled, the compressed files are not uncompressed.

Oracle ACFS Compression Management

- Compressed files are associated with a compression unit and the compression algorithm operates on this unit.
  - For database files, the size of the unit is made equal to the database block size for maximum efficiency.
  - For non-database files, the unit size is currently 32 kilobytes.
- The compression state of a file system can be toggled with `acfsutil compress on` and `acfsutil compress off`.
- The `acfsutil compress copy` command copies and compresses a file or files in a directory.
- To display the compression state and effectiveness of the compression operation, use `acfsutil compress info`.
- `acfsutil info fs` and `acfsutil info file` have been enhanced to report on Oracle ACFS compression status.

ORACLE

Compressed files are associated with a compression unit and the compression algorithm operates on this unit. For database files, the size of the unit is made equal to the database block size for maximum efficiency. For non-database files, the unit size is currently 32 kB. When copying data files to an Oracle ACFS file system that is enabled for compression, use the `acfsutil compress copy` command to ensure that the proper compression unit size is maintained for correct functionality. `lzo` is the default compression algorithm and the only compression algorithm currently supported.

The `acfsutil compress` command sets and resets the compression state of a file system with `acfsutil compress on` and `acfsutil compress off`. To display the compression state and effectiveness of a compression operation, use the `acfsutil compress info` file command. The `acfsutil info fs` and `acfsutil info file` commands have been enhanced to report on Oracle ACFS compression status.

Compressed files consume less disk space than non-compressed files. However, for applications that use the file, the size reported is equal to the uncompressed file size, not the smaller compressed size. Some utilities, such as `ls -l`, report the uncompressed size of the file. Utilities such as `du`, `acfsutil compress info`, and `acfsutil info file` report the actual disk allocation of the compressed file.

ACFS Compression Commands

acfsutil compress on|off

```
$ acfsutil compress on -h
Usage: acfsutil [-h] compress on [-a <algorithm_name>] <mount_point>
                              - Set default compression algorithm
                                Currently only 'lzo' available
                <mount_point> - Enable compression on volume

$ /sbin/acfsutil compress on /u01/acfsmounts/my_fs
```

```
$ acfsutil compress off -h
Usage: acfsutil [-h] compress off <mount_point> - Disable
    compression on volume

$ /sbin/acfsutil compress off /u01/acfsmounts/my_fs
```


The `acfsutil compress on` command enables Oracle ACFS compression for an ACFS file system. When `acfsutil compress on` is run, the following apply:

- Files created after compression is enabled are compressed by default.
- Any uncompressed files in the file system remain uncompressed.

The `acfsutil compress off` command disables Oracle ACFS compression. When `acfsutil compress off` is run, the following apply:

- Files that are created after compression is disabled are created uncompressed.
- Any compressed files in the file system remain compressed.

ACFS Compression Commands

acfsutil compress copy

```
$ acfsutil compress copy -h
Usage: acfsutil [-h] compress copy  [-v] { -c <size> <source> | [-n] {-r
     <source> | <source> [...]} } <target>
                               - Copy a file or files onto an ACFS
                                 file system that has compression enabled
                 [-v]          - Verbose
                 {-c <size> <source> }  - Copy a single file and use a
                                          compression unit size for
                                          the copy
                                 <size> : compression unit size in
                                          bytes as nnn[K]
                 [-n]          - copies will not be compressed
                 {-r <source>} - recursively copy files and directories
                                 starting from a single directory
                 {<source> [...]} - files to copy
                 <target>      - directory to copy into

$ /sbin/acfsutil compress copy -v -f my_file1 my_file2
    /my_target_directory
```


The `acfsutil compress copy` command compresses and copies a file to a specified location. Only files that are recognized as Oracle Database files that are normally compressed are compressed during the copy operation. With these files, the copy is compressed by using the compression unit size that is stored in the Oracle ACFS metadata if it exists; otherwise, the command uses the DB block size stored in the file.

If the file is recognized as a database file, but the DB block size that is stored in the file is not valid as a compression unit size and the Oracle ACFS metadata is not available, the copy is not compressed. If the file is not recognized as a database file, the copy is not compressed.

The command does not copy over existing files unless the `-f` option is used. If `-f` is not used, the copy operation terminates when the first existing target is found. The command does not copy to an Oracle ACFS file system that is capable of supporting compression, if compression is disabled, unless the `-n` option is specified.

ACFS Compression Commands

`acfsutil compress info`

```
$ acfsutil compress info -h
Usage: acfsutil [-h] compress info <pathname>     - Show compression
    information for a regular file

$ /sbin/acfsutil compress info /my_mount_point/testfile1
Compression Unit size: 8192
Disk storage used: ( 189.89 MB )
Disk storage saved: ( 810.20 MB )
Storage used is 18% of what the uncompressed file would use
```

ORACLE

The `acfsutil compress info` command displays information about compressed files.

Oracle ACFS Sparse Files

- Oracle ACFS provides support for sparse files through updates to `acfsutil` commands.
- Sparse files benefit NFS client writes commonly received out of order by the NFS server and the ACFS file system.
- Usually, when an application writes beyond the end of file (EoF):
  - Storage is allocated and zeros inserted for the region of the file beyond the old EoF and the beginning of the new write
- With sparse files, there is a hole in the file instead of zeros:
  - Improves NFS performance
  - Improves the performance and disk utilization of other applications that perform this type of writing intentionally
- The `COMPATIBLE.ADVM` attribute must be set to 12.2 or later.

ORACLE

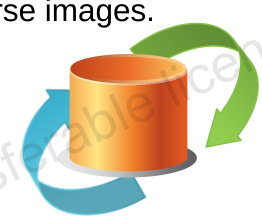
Oracle ACFS provides support for sparse files through updates to `acfsutil` commands. Oracle ACFS sparse files greatly improve NFS client write operations, which are commonly received out of order by the NFS server and the associated Oracle ACFS file system.

Usually, when an application writes beyond the end of file, storage is allocated and zeros inserted for the region of the file beyond the old end of file and the beginning of the new write. With this feature, there is a hole in the file instead of zeros, which improves NFS performance, and also the performance and disk utilization of other applications that perform this type of writing intentionally. To use this feature, ADVM disk group compatibility must be set to 12.2 or later.

Sparse files also benefit the creation and management or use of OVM images. The sparse file approach reduces the overhead for time required to create an image file—no requirement to allocate and zero-fill unused file storage—and for space consumption—no requirement to allocate storage for unused or empty space.

Oracle ACFS Loopback Devices

- With the release of Oracle Database 12.2, ACFS supports the loopback functionality.
- An ACFS loopback device is an operating system pseudo-device that allows an ACFS file to be accessed as a block device.
- This is intended to be used with Oracle VMs in support of the OVM images, templates, and vdisks created in ACFS file systems.
- This functionality provides performance gains against NFS with sparse or non-sparse files.
- In addition, this feature also adds support for loopback direct IO on sparse images.



Oracle ACFS supports the loopback functionality. An ACFS loopback device is an operating system pseudo-device that enables an ACFS file to be accessed as a block device. This functionality is intended to be used with Oracle Virtual Machines in support of the OVM images, templates, and virtual disks (vdisks) that are created in ACFS file systems and presented through ACFS loopback devices.

This functionality provides performance gains against NFS with sparse or non-sparse files. In addition, this feature also adds support for loopback direct IO on sparse images.

Quiz

Oracle ACFS Defragger allows live Oracle ACFS file systems to be defragged on demand.

a. True

b. False

Answer: a

This statement is true.

Summary

In this lesson, you should have learned how to:

- Understand the following ACFS/ADVM enhancements in 12.2
  - Oracle ACFS Automatic Resize
  - Oracle ACFS Scrubbing
  - Oracle ACFS Defragger
  - Oracle ACFS Metadata Collection
  - Oracle Metadata Acceleration
  - 4K Sector Support for Oracle ACFS and Metadata
  - Oracle ACFS Compression
  - Oracle ACFS Spare Files
  - Oracle ACFS Loopback Devices

Practice 9 Overview: ACFS/ADVM Enhancements

This practice covers the following topics:

- 9-1: ACFS Automatic Resizing
- 9-2: ACFS Defragmentation
- 9-3: ACFS Compression

A

Oracle ASM Advanced Topics

Objectives

After completing this lesson, you should be able to:

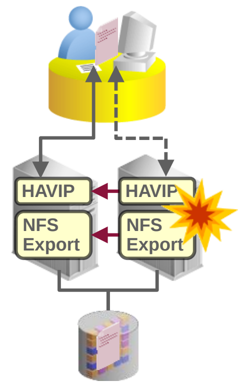
- Configure High Availability NFS
- Describe how to configure ASM Filter Driver
- Upgrade an existing database to use ASM Filter Driver and see its effect
- Configure and manage ACFS auditing
- Implement ACFS encryption
- Configure and manage ACFS replication
- Configuring and manage ACFS Snapshot-Based replication

ORACLE

High Availability NFS

High Availability NFS (HANFS) provides uninterrupted NFS service:

- Exported file systems are exposed by using Highly Available Virtual Ips (HAVIPs).
- Oracle Clusterware manages the NFS exports and HAVIPs.
  - Services are automatically migrated if the current node fails.



ORACLE

With Oracle Database 12*c*, ACFS includes High Availability NFS (HANFS). HANFS provides uninterrupted service of NFS exported paths by exposing NFS exports on Highly Available Virtual IPs (HAVIPs). Oracle Clusterware agents are used to ensure that the HAVIPs and NFS exports are always online. If a cluster node fails, the HAVIPs and NFS exports are automatically migrated to a surviving node.

HANFS works in conjunction with NFS version 2 and NFS version 3.

Configuring High Availability NFS

- Ensure that NFS is running.
- After creating an ACFS file system:
  - Register the ACFS file system as a cluster resource:

```
# srvctl add filesystem -d /dev/asm/vol1-201 \
> -m /mnt/acfsmounts/acfs1
```

  - Mount the ACFS file system on all cluster nodes:

```
# srvctl start filesystem -device /dev/asm/vol1-201
```

  - Register a new HAVIP resource:

```
# srvctl add havip -address c01vip -id havip1
```

  - Register the ACFS file system export:

```
# srvctl add exportfs -id havip1 -path /mnt/acfsmounts/acfs1 \
> -name export1 -options rw -clients *.example.com
```

  - Export the file system:

```
# srvctl start exportfs -name export1
```

ORACLE

Before configuring HANFS, ensure that an NFS server is running on the required cluster nodes and that you have created the ACFS file systems that you want to expose with HANFS.

HANFS requires the underlying ACFS file system to be registered as a cluster resource and mounted on multiple cluster nodes. This can be achieved by using the `srvctl add filesystem` and `srvctl start filesystem` commands, as shown in the examples in the slide. The key parameter for the `srvctl ... filesystem` commands is the device that is associated with the ACFS file system. Administrators can view the device that is associated with an ACFS file system by using the `volinfo --all` command in the ASMCMD command-line utility.

Also, a HAVIP resource must be created. The key parameter for the HAVIP resource is the address, which is specified by using a host name or IP address. The corresponding IP address must be a single static address (no dynamic host configuration protocol [DHCP] or round-robin domain name server [DNS] resolution), which is not currently in use and is on the same subnet as the existing node VIPs.

After the HAVIP is defined, an EXPORTFS resource must be created. To create the EXPORTFS resource, you must specify the HAVIP resource that will be used to export the file system, the path of the ACFS file system being exported, and a name that is used to identify the resource. You can also specify other NFS options and allowed clients.

After all the resources are in place, the file system can be exported by using the `srvctl start exportfs` command. Exporting the file system automatically starts the associated HAVIP.

Highly Available NAS Enhancements

- Oracle ACFS NAS Maximum Availability eXtensions provide HA extensions for common NAS protocols such as NFS and SMB.
- When using these extensions, the protocol runs in HA mode, enabling it to move between nodes in a RAC cluster.
  - This provides a way to address a single point of failure for a given protocol, so that if at least one node of the cluster is available, the protocol is available.
- In addition to providing for HA, the extensions allow integration with common NAS protocols and the Oracle ACFS stack.
- The Oracle ACFS *N|MAX* functionality adds value to existing OS NAS protocol implementations, but does not replace them.

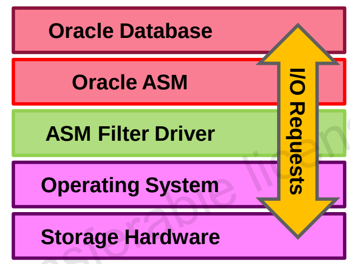
ORACLE

Oracle ACFS NAS Maximum Availability eXtensions (Oracle ACFS N|MAX) includes a set of extensions that provide High Availability for common NAS protocols, such as NFS and SMB. When using these extensions, the protocol in question runs in HA mode, enabling it to move between nodes in an Oracle RAC cluster. This functionality provides a way to address a single point of failure for a given protocol, so that if at least one node of the cluster is available, the protocol is available. In addition to providing for high availability, the extensions provide for integration with common NAS protocols and the Oracle ACFS stack, enabling administrators to easily use these protocols without creating additional infrastructure. Note that the Oracle ACFS NAS Maximum Availability eXtensions functionality adds value to existing OS NAS protocol implementations, but does not replace them.

ASM Filter Driver: Overview

Key Capabilities:

- Rejection of non-Oracle I/O
  - Stops OS utilities from overwriting ASM disks
  - Protects database files
- Reduction of OS resource usage
  - Fewer open file descriptors
- Faster node recovery
  - Restarting Oracle Clusterware instead of restarting the node



The Oracle ASM Filter Driver (ASMFD) is a kernel module that resides in the I/O path of the Oracle ASM disks. Logically, ASMFD provides an interface between Oracle binaries and the underlying operating environment, which includes the storage hardware interfaces. Following are descriptions of the key capabilities of ASMFD:

- **Reject Non-Oracle I/O:** As a manager of Oracle storage, ASM is exposed to OS capabilities when dealing with storage devices. In particular, non-Oracle commands have the ability to overwrite the contents of ASM disks, which may lead to irrecoverable data loss. ASMFD allows writes only by using an Oracle-specific interface and prevents non-Oracle applications from writing to ASM disks. This protects ASM from accidental corruption.
- **Reduce OS Resource Usage:** An ASM instance contains numerous processes or threads on Windows. Without ASMFD, each process that is I/O-capable needs to have its own dedicated open file descriptor for each disk. When thousands of processes access hundreds of disks, the ensuing explosion of file descriptors leads to considerable OS resource consumption. ASMFD exposes a portal device that can be used for all I/O on a particular host. The same portal device can be shared by all the processes that are associated with multiple database instances. By using ASMFD, you drastically reduce the required number of open file descriptors.
- **Faster Node Recovery:** With the current implementation of Oracle Clusterware, `init.d` scripts restart the node when CSS is not functioning correctly. Using this mechanism, nodes are fenced to ensure integrity of the rest of the cluster. Although effective, this solution is costly because of the time required to restart the node and restart all the required processes. ASMFD allows Oracle Clusterware to perform node-level fencing without a restart. So with ASMFD, it is possible to achieve the same result by restarting the Oracle software stack instead of restarting the entire node. This process is just as effective, but far quicker.

Note that ASMFD is not available in the Beta 2 release of Oracle Database 12*c*.

Configuring the ASM Filter Driver

- ASMFD is installed during ASM installation.
  - Automatically enabled by default
- Further configuration and monitoring is performed by using the ADFTOOL utility:
  - Provision a disk:

```
$ adftool add /dev/dsk1 disk1
```

  - Enable or disable ASMFD for a disk:

```
$ adftool enablefilter /dev/dsk1
```

```
$ adftool disablefilter /dev/dsk1
```

  - Remove a disk:

```
$ adftool delete disk1
```

  - List the managed disks:

```
$ adftool getdevlist
```

ORACLE

ASMFD is installed during ASM installation, and is automatically enabled by default. However, to use ASMFD, storage devices must be registered with the driver.

ASMFD is managed by using the ADFTOOL utility. A summary of the ADFTOOL commands is as follows:

- `adftool add <path> <label>` provisions the disk identified by the `<path>` parameter for Oracle ASM future usage.
- `adftool delete [-f] <label>` clears the Oracle ASM label from the specified storage device. This operation is required to free up an ASM provisioned disk for other applications. The `-f` option forces the action.
- `adftool getdevlist [filter|stamped|discovered]` lists the devices that are managed by ASMFD.
- `adftool enablefilter|disablefilter <path>` enables or disables ASMFD for the specified storage `<path>`. The `<path>` must refer to a disk that was previously discovered and stamped by Oracle ASM.
- `adftool stop` frees all the resources used by ASMFD, and unloads the driver from the system.

Labeling Disks for ASM Filter Driver

- Disks must be labeled to enable management by ASMFD.
- To label disks:
  1. Prepare for disk labeling:
     – You must shut down Oracle Clusterware on all cluster nodes to label disks in a disk group that contains the OCR or voting files.
     – You must dismount the associated disk group to label disks in a disk group that does NOT contain the OCR or voting files.
  2. Label disks:
     ```
     $ asmcmd afd_label <Label Name> <Device> [--migrate]
     ```
     – Label disks only once on any cluster node.
  3. Discover the labeled disks:
     ```
     $ asmcmd afd_scan
     ```
     – Labeled disks must be discovered on every cluster node.

ORACLE

To enable ASMFD management, each disk device must be labeled by ASMFD.

1. If the disk currently belongs to a disk group, before labeling, the disk group must be dismounted across the entire cluster. If the disk group contains the Oracle Cluster Registry (OCR) or voting files, you must shut down Oracle Clusterware on all the cluster nodes before labeling the disk.
2. You can use the `asmcmd afd_label` command or the `ALTER SYSTEM LABEL SET` SQL statement to label each disk device. If the disk currently belongs to a disk group, you must use the `migrate` option. Disk labeling is a one-time operation that can be performed on any cluster node.
3. Labeled disks must be discovered on every cluster node. You can use the `asmcmd afd_scan` command or the `ALTER SYSTEM LABEL SCAN` SQL statement to perform a scan on each cluster node. You do not need to perform an ASMFD disk scan on the node where you labeled your disks. For example, if you labeled all your disks by using one cluster node, you would need to perform a scan on all the other cluster nodes. However, if you labeled different disks by using different cluster nodes, you would need to perform a scan on all the nodes to ensure that all the disks were discovered on all the nodes.

Migrating from ASMLib to ASMFD

1. Reset the ASM disk string to an empty string:
```
$ asmcmd dsset ''
```
2. Stop Oracle Clusterware.
3. Configure ASMFD:
```
# asmcmd afd_configure
```
   – This must be performed by the `root` OS user on every node.
   – ASMlib labels are automatically converted to ASMFD.
4. Discover the labeled disks:
```
$ asmcmd afd_scan '/dev/xvd*'
```
5. Restart Oracle Clusterware.
6. Set the ASMFD disk string:
```
$ asmcmd afd_dsset '/dev/xvd*'
```

ORACLE

To migrate ASMlib disks to ASMFD, users must first reset the ASM disk string to an empty string. An empty ASM disk string causes ASM to scan for disks in a number of default locations, including the default ASMlib path `ORCL:*`, so your ASMlib disks will remain visible to ASM. However, this setting is required by the ASMFD configuration routines, which will otherwise raise an error if ASMlib disks are found in conjunction with a nonempty ASM disk string.

After the ASM disk string is reset, the remaining tasks must be performed on each cluster node. Note that the ASMFD configuration process (step 3) automatically disables ASMlib and converts all ASMlib disk labels to ASMFD labels. Following this step, ensure that the newly relabeled disks are visible to ASMFD by executing a disk scan (step 4). At this point, the ASMFD disk string is not yet set; therefore, you must provide the appropriate device disk string to the `asmcmd afd_scan` command. After you restart Oracle Clusterware (step 5), you should set the ASMFD disk string (step 6) to ensure that the disk devices are automatically discovered by ASMFD from then on. Note that the ASMFD disk string must be set on each node.

Unlabeling Disks and Deconfiguring ASMFD

- Unlabeling a disk:

```
# asmcmd afd_unlabel <Label Name> [-f]
```

  – Can happen at any time if the disk is not in a disk group
- Deconfiguring ASMFD:

```
# asmcmd afd_deconfigure
```

  – Must be performed by the `root` OS user
  – Must be performed separately on each node
  – Must stop CRS

The ASMCMD utility provides commands for unlabeling disks and deconfiguring ASMFD.

To unlabel a disk, use the `asmcmd afd_unlabel` command and supply the label name for the disk that is being unlabeled. You can unlabel a disk at any time if it does not belong to an ASM disk group. The most likely use case occurs when you return a disk device to a pool of non-Oracle storage. If you want to unlabel a disk that belongs to an ASM disk group, you must at least unmount the disk group. If the disk group in question contains the OCR or voting disks, you must also stop Oracle Clusterware on all the cluster nodes before you can unlabel the disk.

If you want to completely deconfigure ASMFD for any reason, you can use the `asmcmd afd_deconfigure` command. This command must be performed separately on each cluster node by the `root` OS user and Oracle Clusterware must be shut down during this process.

ACFS Auditing

With Oracle Database 12*c*, ACFS introduces a general audit framework for file systems:

- A separate audit trail can be defined for each file system.
- It enables separation of duties to be enforced.
- A collector for Oracle Audit Vault is also available.
  - Audit Vault provides secure offline audit trail storage with built-in analysis and reporting tools.
- Consistent functionality is provided across all platforms that are supported by ACFS.

ORACLE

In previous versions, ACFS provided some support for auditing; however, there was no unified management framework and no separation of duties to help protect the integrity of the audit trail.

With Oracle Database 12*c*, ACFS introduces a general auditing framework for ACFS file systems. This auditing framework can produce a separate audit trail for each file system, and enforce separation of duties regarding the management and review of the audit trail.

Along with the generation of a file system audit trail, a collector for Oracle Audit Vault has been developed. This collector is separate from ACFS, but provides a way for ACFS audit data to be integrated into Oracle Audit Vault.

Although the concept of file system auditing is not new, ACFS auditing delivers the following set of advanced capabilities:

- It provides the same functionality and management interfaces across all the platforms that ACFS supports.
- It enforces separation of duties for audit trail management and review.
- It integrates with Oracle Audit Vault, which provides a secure offline store for audit information, along with analysis and reporting capabilities.

ACFS Audit Trail Files

- Audit files are located at:

```
Mount_Point/.Security/audit/acfs-audit-<FSID>-<Hostname>.log
```

- Audit files are secured by using permissions that enforce separation of duties.
  - Audit managers can manage the audit trail.
  - Auditors can view but cannot manage.
  - Neither role can truncate, overwrite, or delete the audit trail.
- When audit files are full, they are automatically archived.
  - Files that are not full can be manually archived.
- Archive files are located at:

```
Mount_Point/.Security/audit/acfs-audit-<FSID>-<Hostname>.log.bak
```

- Active audit files should not be interrogated.
  - Archive first, and then interrogate the archive.

ORACLE

The ACFS audit trail is contained in a set of files inside ACFS. The audit trail files generated by ACFS auditing are designed for:

- Manual review by an ACFS auditor by using text-based viewing tools
- Integration into Oracle Audit Vault
- Integration into third-party products that can parse and import the audit data

After auditing is enabled, audit files are written under `mount_point/.Security/audit`. Each host in the ACFS cluster writes to a separate audit file. This avoids potential complications that are associated with multiple hosts attempting to use the same file.

The audit files are secured by using permissions that enforce separation of duties. That is, audit managers can perform management functions, such as archiving the audit files, but cannot mark the archived files as read. Auditors can view the contents of audit files and mark them as read, but they cannot perform management functions. Note that audit managers and auditors cannot truncate, overwrite, or delete the audit trail.

When audit files reach 10 MB, they are considered full and are automatically archived. When a file is archived, it is closed and `.bak` is appended to the file name. The next audit record is written to a new audit file, enabling auditing to continue without interruption. Files that are not full can be manually archived by using the `acfsutil audit archive` command.

Note that active audit files should not be interrogated because it could interrupt auditing or result in the loss of auditing data. Rather, an archive should be created and interrogated instead of the active audit file.

Audit Trail Contents

Sample audit file:

```
ACFS Audit Version: 1.0
Encoding: UTF-8
```
Header

```
Event: ACFS_CMDRULE_WRITE
Description: A user attempted to write to a realm protected file.
Product: ACFS_SECURITY
Timestamp: 2/21/2012 08:23:01 UTC
User: 102
Group: 102
Process: 4567
Host: host1
File: /my_mount_point1/hr_data/payroll
Evaluation Result: ACFS_REALM_AUTH
Realm: myPayrollRealm
Application: vi
```
File Access Event

```
Event: ACFS_SEC_PREPARE
Description: A user prepared a device for ACFS Security.
Product: ACFS_SECURITY
Timestamp: 2/23/2012 09:14:10 UTC
User: 1042
Group: 1823
Process: 8901
Host: host1
Command Line: acfsutil sec prepare -m /my_mount_point2
```
Privilege Use Event

The ACFS audit trail consists of a set of audit records. Each audit record represents a single event.

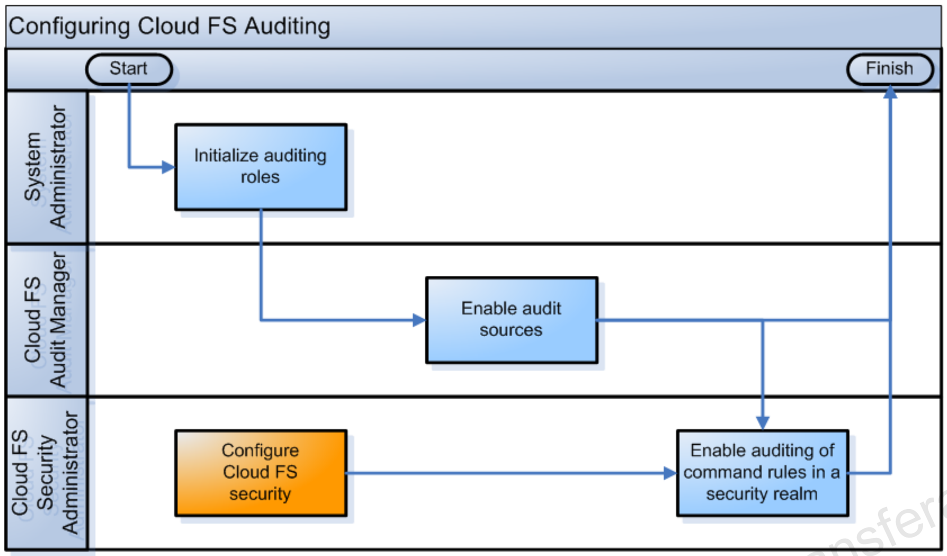
The audit trail has a brief header at the beginning of each file. The header identifies the version of the audit file format and the character encoding for the audit file.

Following the header, the audit file consists of audit records. There are several different types of audit records, each of which represents a unique type of event and contains different information that is relevant to the event. The types of events are:

- File access events
- Privilege use events
- Authentication failures or insufficient privileges events

Each record is written to the audit trail as a set of field names and values. Each field and value pair is separated by a colon and followed by an end-of-line character. The combination of audit record fields entered in the audit trail depends on the event type. Refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)* for a complete listing of all the audit events and audit record fields.

Configuring ACFS Auditing



ORACLE

The diagram in the slide illustrates the process for configuring ACFS auditing. It outlines the key process tasks and who is required to perform them.

Note that the task of configuring ACFS security is separate from configuring ACFS auditing, but is mentioned here because it is an important prerequisite. Note also that the diagram does not mention other prerequisite tasks, such as initially configuring ACFS and creating ACFS file systems.

Further detail regarding the steps for configuring ACFS auditing are provided on the following pages.

Initializing Auditing Roles and Enabling Audit Sources

- Set up the required roles for auditing:
  - Required task
  - Performed by the system administrator
  - Command syntax:

```
# acfsutil audit init –M Audit_Manager_Group –A Auditor_Group
```

  - Groups cannot be changed after initialization.
- Enable auditing on a specified file system:
  - Required task
  - Performed by an ACFS audit manager
  - Command syntax:

```
$ acfsutil audit enable –m Mount_Point -s [sec | encr]
```


The `acfsutil audit init` command must be run by the system administrator. The command sets up the required roles for auditing and must be run before any type of auditing can be enabled on a file system. After initialization, you cannot choose a different OS group for either the ACFS audit manager or ACFS auditor. Because of this, it is recommended that specific OS groups should be created for these roles.

The `acfsutil audit enable` command enables auditing on a specified file system. Only an audit manager can run this command. In addition to specifying the file system, the audit manager must specify whether to enable auditing for ACFS security (`–s sec`) or ACFS encryption (`–s encr`). Auditing can be enabled for both ACFS security and ACFS encryption by running the `acfsutil audit enable` command twice.

Enabling Auditing of Command Rules in a Security Realm

Enable auditing of specific command rules in an ACFS security realm.

- Optional task
- Performed by an ACFS security administrator
- Command syntax:

```
$ acfsutil sec realm audit enable Realm –m Mount_Point
  [-l Command_Rule,Command_Rule,...] [-a] [-v [-u]]
```

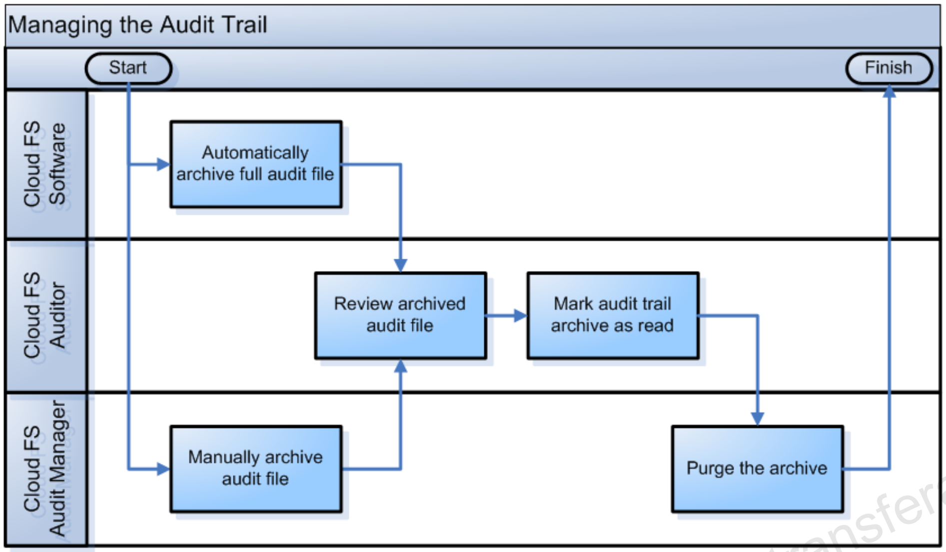

In addition to the core auditing that is enabled by the `acfsutil audit enable` command, auditing of specific command rules in an ACFS security realm can be enabled by using the `acfsutil sec realm audit enable` command. Note that only the security administrator, not the audit manager, can run this command. Following is a description of the options that are available with the `acfsutil sec realm audit enable` command:

`Realm`	Specifies the security realm name
`-m` *`Mount_Point`*	Specifies the directory where the file system is mounted
`-l` *`Command_Rule`*	Specifies the command rules that are audited. If it is not specified, all command rules associated with the realm are audited.
`-a`	Specifies audit realm authorizations
`-v [-u]`	Specifies audit realm violations. If `-u` is specified, only realm violations by users who are members of a realm are audited.

Auditing of command rules in a security realm builds on the realm-based security capabilities that are already present in earlier releases of ACFS. For more information about ACFS security, including realms and command rules, refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)*.

Note that ACFS security realms and command rules must be configured before auditing of command rules in a security realm can be enabled.

Managing the Audit Trail



The diagram in the slide illustrates the process for managing the ACFS audit trail in the absence of Oracle Audit Vault. It outlines the key process tasks and who performs them.

The diagram represents one iteration of what is an endless loop as archive files fill up over and over again.

If Oracle Audit Vault is implemented, the Oracle Audit Vault collector automatically consumes the archived files and marks them as read.

Further detail regarding the steps for managing the audit trail are provided on the following pages.

Archiving Audit Files

- Audit files are automatically archived when they reach the predefined maximum size of 10 MB.
- Audit files can also be manually archived by an ACFS audit manager.
  - Immediate review of recent audit data enabled
  - Command syntax:

```
$ acfsutil audit archive –m Mount_Point
```


Audit files are automatically archived when they reach the predefined maximum size of 10 MB. In addition, ACFS audit managers can manually archive audit files by using the `acfsutil audit archive` command. This enables immediate review of recent audit data in the archive while the system continues to record audit data in the active file.

Reviewing Audit Files

- Without Oracle Audit Vault:
  - Auditors can review archived audit files by using any tools.
    - Can back up the archive or copy audit data into another file, if necessary
  - Archived audit files should be marked as read when they are no longer required.
    - Indicates that it is safe to purge the archived files
    - Command syntax:

```
$ acfsutil audit read -m Mount_Point
```

- With Oracle Audit Vault:
  - Archived audit files are automatically imported into Oracle Audit Vault.
    - Automatically marked as read after successful import
  - Auditors should use Audit Vault tools to review the audit trail.

ORACLE

By default, auditors can review the audit trail by using any tools to read or search against the archived audit files. Auditors are free to back up the archive or copy audit data into another file, if necessary. Remember that active audit files should not be interrogated because it could interrupt auditing or result in the loss of audit data.

When they are no longer required, archived audit files should be marked as read to indicate that it is safe to purge them. Use the `acfsutil audit read` command to mark the files as read.

If Oracle Audit Vault is implemented, archived audit files are automatically imported into Oracle Audit Vault. After they are successfully imported, the archived audit files are automatically marked as read. In this case, auditors should use the Audit Vault tools to analyze and review the audit trail, rather than accessing the audit files directly.

Purging Audit Files

Purging removes archived the audit files that have been marked as reviewed:

- Must be performed by an audit manager
- Is important because you cannot archive the current audit file before the previous archive is purged
- Command syntax:

```
$ acfsutil audit purge –m Mount_Point [-f]
```

ORACLE

Purging removes the archived audit files that have been marked as reviewed.

Purging is important because you cannot archive the current audit file before the previous archive is purged.

Purging is performed by the audit manager by using the `acfsutil audit purge` command. This command has one optional argument (`-f`) that forces purging even if the auditor has not marked the archived audit files as read. Forced purging can result in the loss of audit data and should generally not be performed.

ACFS Encryption

- ACFS encryption enables you to encrypt data stored in an ACFS file system.
- ACFS encryption protects the data in an ACFS file system to prevent unauthorized data use in the case of data loss or theft.
- Both encrypted and non-encrypted files can exist in the same Oracle ACFS file system.
- ACFS encryption provides two types of encryption keys:
  – File Encryption Key: Used to encrypt the data in the file
  – Volume Encryption Key: A key for a file system that is used to encrypt the file encryption keys
- Back up the OCR after creating an encryption key to ensure that there is a backup containing all the volume encryption keys for the file system.

ORACLE

Oracle ACFS encryption enables you to encrypt the data stored on disk (data-at-rest). The encryption feature protects the data in an Oracle ACFS file system to prevent unauthorized use of data in the case of data loss or theft. Both encrypted and non-encrypted files can exist in the same Oracle ACFS file system.

Some encryption functionality require system administrator privileges, including commands for initiating, setting, and reconfiguring encryption. System administrators and Oracle ACFS security administrators can initiate encryption operations. Also, unprivileged users can initiate encryption for the files they own.

Oracle ACFS encryption provides two type of encryption keys:

- **File Encryption Key:** This is a key for a file and is used to encrypt the data in the file.
- **Volume Encryption Key:** This is a key for a file system and is used to encrypt the file encryption keys.

You must first create the encryption key store, and then specify the file system–level encryption parameters and identify the directories. No extra steps are required for a user to read encrypted files if the user has the appropriate privileges for accessing file data.

You should back up the Oracle Cluster Registry (OCR) after creating or updating an encryption key to ensure that there is an OCR backup that contains all the volume encryption keys (VEKs) for the file system.

ACFS Encryption

- An ACFS security administrator can manage encryption parameters on a per-realm basis.
- After a directory has been added to a realm, the files that are created in the directory inherit the realm-level encryption parameters.
- Auditing and diagnostic data are logged for ACFS encryption. Logs are written to the following files:
  - `mount_point/.Security/encryption/logs/encr-hostname_fsid.log`
  - `GRID_HOME/log/hostname/acfs/security/acfssec.log`
- Compatibility attributes for ASM and ADVM must be set to:
  - 11.2.0.2 or later for Linux
  - 11.2.0.3 or later for Windows
- Encryption information for ACFS file systems is displayed in the `V$ASM_ACFS_ENCRYPTION_INFO` view.

ORACLE

An Oracle ACFS security administrator can manage encryption parameters on a per-realm basis. After a file is placed under realm security, file-level encryption operations are not allowed on that file. Even if the realm security allows the file owner or the root user to open the file, file-level encryption operations are blocked. Encryption of realm-protected files is managed entirely by the Oracle ACFS security administrator, who can enable and disable encryption for files at a security realm level.

After a directory is added to a security realm, all the files that are created in the directory inherit the realm-level encryption parameters, not the directory or file system–level parameters. When a file is removed from its last security realm, the file is encrypted or decrypted to match the file system–level encryption status. The file is not re-encrypted to match the file system–level parameters if it has been encrypted with security realm parameters.

A system administrator cannot rekey realm-secured files at the file system or file level. To ensure that all realm-secured files are encrypted with the most recent volume encryption key (VEK), you must first remove encryption from all realms, and then re-enable encryption. This action re-encrypts all files with the most recent VEK.

Auditing and diagnostic data are logged for Oracle ACFS encryption. The log files include information such as the `acfsutil` commands that have been run, the use of security or system administrator privileges, and run-time failures.

Logs are written to the following files:

- *mount_point*/.Security/encryption/logs/encr-*hostname_fsid*.log

  The directory is created with the acfsutil encr set command and protected by Oracle ACFS security if security is enabled.

- *GRID_HOME*/log/*hostname*/acfs/security/acfssec.log

  The messages that are logged to this file are for commands that are not associated with a specific file system, such as acfsutil encr init. The directory is created during installation and is owned by the root user.

When an active log file grows to a predefined maximum size (10 MB), the file is automatically moved to log_file_name.bak, the administrator is notified, and logging continues to the regular log file name. When the administrator is notified, the administrator must archive and remove the log_file_name.bak file. If an active log file grows to the maximum size and the log_file_name.bak file exists, logging stops until the backup file is removed. After the backup log file is removed, logging restarts automatically.

To use ACFS encryption on Linux, the disk group compatibility attributes for ASM and ADVM should be set to 11.2.0.2 or later. The disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or later on Linux for the following cases:

- If encryption is configured for the first time on Oracle ASM 11*g* Release 2 (11.2.0.3)
- If encryption parameters must be changed or a new volume encryption key must be created following a software upgrade to Oracle ASM 11*g* Release 2 (11.2.0.3)

To use Oracle ACFS encryption functionality on Windows, the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or later.

Encryption information for Oracle ACFS file systems is displayed in the V$ASM_ACFS_ENCRYPTION_INFO view.

Encrypting ACFS File Systems

The basic steps to manage encryption are as follows:

1. Initialize encryption.

```
# /sbin/acfsutil encr init
```

2. Set encryption parameters:

```
# /sbin/acfsutil encr set -a AES -k 128 –m \
/acfsmnts/acfs1/
```

3. Enable encryption for directories and files:

```
# /sbin/acfsutil encr on -r /acfsmnts/acfs1/medrecords \
-m /acfsmnts/acfs1/
```

4. Display encryption information.

```
# /sbin/acfsutil encr info -m /acfsmnts/acfs1/ -r
/acfsmnts/acfs1/medrecords
```

ORACLE

The basic steps to manage encryption are as follows:

1. Initialize encryption. Run the `acfsutil encr init` command to initialize encryption and create the storage that is necessary for the encryption keys. This command must be run once for each cluster on which encryption is set up. For example, to initialize encryption for a cluster:

   ```
   # /sbin/acfsutil encr init
   ```

   This must be run before any other encryption command and requires root privileges to run.
2. Set encryption parameters. Run `acfsutil encr` set to set the encryption parameters for the entire ACFS file system. For example, the following command sets the AES encryption algorithm and a key length of 128 for a file system that is mounted on /`acfsmnts/acfs1`:

   ```
   # /sbin/acfsutil encr set -a AES -k 128 -m /acfsmnts/acfs1/
   ```

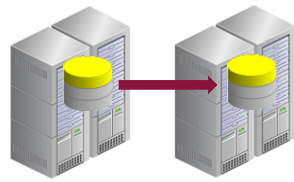
   The preceding command also transparently generates a volume encryption key, which is stored in the key store that was previously configured with the `acfsutil encr init` command.
3. Enable encryption. Run acfsutil encr on to enable encryption for directories and files. For example, the following command enables encryption recursively on all the files in the /`acfsmnts/acfs1/medrecords` directory.

   ```
   # /sbin/acfsutil encr on -r /acfsmnts/acfs1/medrecords -m
   /acfsmnts/acfs1/
   ```

   Users that have permission to access the files in the medrecords directory can still read the decrypted files. Run `acfsutil encr` info to display encryption information.

ACFS Replication

- ACFS replication enables replication of ACFS file systems to a remote site, thus providing disaster-recovery capabilities.
- ACFS replication can be configured only for Oracle RAC systems.
- ACFS replication captures file system changes to replication logs.
  - The logs are transported to the site that is hosting the associated standby file system.
  - Background processes read the logs and apply the changes recorded in the logs to the standby file system.
  - The logs are deleted at the primary and standby sites after the changes have been applied to the standby file system.
- There must be enough disk space on the primary and standby sites to contain the replication logs.



Oracle ACFS replication enables replication of Oracle ACFS file systems across the network to a remote site, thus providing disaster-recovery capability for the file system. Oracle ACFS replication can be configured only for Oracle RAC systems. The source Oracle ACFS file system of an Oracle ACFS replication is referred to as a primary file system. The target Oracle ACFS file system of an Oracle ACFS replication is referred to as a standby file system.

A site can host both primary and standby file systems. For example, if there are cluster sites A and B, a primary file system that is hosted at site A can be replicated to a standby file system at site B. Also, a primary file system that is hosted at site B can be replicated to a standby file system at site A. However, an ACFS file system cannot be used as a primary and a standby file system.

Oracle ACFS replication captures the file system changes that are written to disk for a primary file system and records the changes in files called replication logs. These logs are transported to the site that hosts the associated standby file system where background processes read the logs and apply the changes recorded in the logs to the standby file system. After the changes recorded in a replication log have been successfully applied to the standby file system, the replication log is deleted from the sites that host the primary and standby file systems.

It is critical that enough disk space is available on both sites hosting the primary and the standby file systems to contain the replication logs.

If the primary file system runs out of space, applications running on the file system may fail because Oracle ACFS cannot create a new replication log to capture the file system changes made by the application. If the standby file system runs out of space, it cannot accept new replication logs from the primary file system and cannot apply those changes to the standby file system. In addition, replication logs accumulate on the primary file system and consume the available disk space.

If the primary file system has less than 2 GB available free disk space, Oracle ACFS attempts to automatically terminate replication on the primary file system. This action prevents further consumption of disk space for replication operations and frees disk space consumed by any replication logs that remain. The auto-terminate process can prevent the primary file system from running out of space in most cases, but it is still possible that the auto-terminate process does not occur quickly enough. Before reaching the 2-GB limit, Oracle ACFS writes warnings about the free space problem in the Oracle Grid Infrastructure home alert log.

You should prevent both the primary file system and the standby file system from running out of space. If either file system runs out of available storage, you should either expand the file system or remove files from the file system to free up space. If the primary file system runs out of space and you decide to free up space by removing files, you should remove only files that are not being replicated because removal of a file that is replicated is captured in a replication log. Another option is to delete any Oracle ACFS snapshots. Because replication logs can accumulate when replication is paused, you should resume replication soon after pausing replication.

ACFS Replication Requirements

- There must be sufficient network bandwidth to support replication between the primary and standby file systems.
- The primary and standby site configuration must allow the standby file system to keep up with the rate of change.
- The standby file system must have sufficient capacity to manage the replication logs that are sent.
- The primary file system must have a minimum size of 4 GB for each node that is mounting the file system.
- The standby file system must be at least 4 GB and sized appropriately for the amount of data that is being replicated.

ORACLE

Before using replication on a file system, ensure that you have checked the following:

- There is sufficient network bandwidth to support replication between the primary and standby file systems.
- The configuration of the sites that host the primary and standby file systems allow the standby file system to keep up with the rate of change on the primary file system.
- The standby file system has sufficient capacity to manage the replication logs.
- There is sufficient storage capacity to hold excess replication logs that might collect on the primary and the standby file systems when the standby file system cannot process replication logs quickly. For example, this situation can occur during network problems or maintenance on the site that is hosting the standby file system.
- The primary file system must have a minimum size of 4 GB for each node that is mounting the file system. The standby file system must have a minimum size of 4 GB and should be sized appropriately for the amount of data being replicated and the space that is necessary for the replication logs that are sent from the primary file system.

Before replicating an ACFS file system, a replication configuration must be established that identifies information such as the site that is hosting the primary and standby file systems, the file system to be replicated, mount point of the file system, and a list of tags if desired.

To use Oracle ACFS replication functionality on Linux, the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.2 or later for the disk groups that contain the primary and standby file systems. To use Oracle ACFS replication functionality on Windows, the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or later. To use Oracle ACFS replication functionality on Solaris or AIX, the disk group compatibility attributes for ASM and ADVM must be set to 12.1 or later.

To configure replication and manage replicated Oracle ACFS file systems, use the `acfsutil repl` command-line functions.

Managing ACFS Replication

The basic steps for managing ACFS replication are:

1. Determine the storage capacity necessary for replication on the sites hosting the primary and standby file systems.
2. Set up usernames, service names, and tags.

```
SQL> CREATE USER primary_admin IDENTIFIED BY primary_passwd;
SQL> GRANT sysasm,sysdba TO primary_admin;
```

```
primary_repl_site=(DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=primary1.example.com)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=primary2.example.com)(PORT=1521))
   (CONNECT_DATA=(SERVICE_NAME=primary_service)))

standby_repl_site=(DESCRIPTION= ...
```

3. Configure the site hosting the standby file system.

```
$ /sbin/acfsutil repl init standby \
  -p primary_admin/primary_passwd@primary_repl_site \
  -c standby_repl_service /standby/repl_data
```

ORACLE

The steps to manage ACFS replication are as follows:

1. Determine the storage capacity that is necessary for replication on the sites that are hosting the primary and standby file systems. The primary file system must have a minimum size of 4 GB for each node that is mounting the file system. The standby file system must have a minimum size of 4 GB and should be sized appropriately for the amount of data that is being replicated and the space that is necessary for the replication logs that are sent from the primary file system. Calculate the replication-related storage requirement for the primary file system, and then use the same size requirement for the standby file system.
2. Within ASM, set up tags, usernames, and service names. Determine the username and password that the sites that host the primary and standby file systems use to connect to the remote ASM instance as the Oracle ASM and DBA administrator. All nodes that have the file system mounted must support this user. The user must have `SYSASM` and `SYSDBA` privileges.

If you want to replicate by using a SCAN VIP, you must update the `REMOTE_LISTENER` initialization parameter in the ASM instance before initializing replication. You can update the parameter in the initialization file or with the `ALTER SYSTEM SQL` statement. For example:

```
SQL> ALTER SYSTEM SET remote_listener='SCAN_NAME:1521' sid='*' scope=both;
```

3. Before replicating an Oracle ACFS file system, configure the site that is hosting the standby file system by performing the following procedures.

Create a new file system of adequate size to hold the replicated files and the associated replication logs from the primary file system. Mount the file system on one node only. Run the `acfsutil repl init standby` command.

This command requires the following configuration information:

- The connect string to be used to connect to the site that is hosting the primary file system as specified by the `-p` option, for example:

  `primary_admin/primary_passwd@primary_repl_site`

  Note: The user `primary_admin` must have `SYSASM` and `SYSDBA` privileges.

- If the standby file system is using a different service name than the primary file system, the `-c` option. This option specifies the service name for the standby file system. For example:

  `standby_repl_service`

- The mount point of the standby file system. For example:

  `/standby/repl_data`

If this command is interrupted for any reason, the user must re-create the file system, mount it on one node only, and rerun the command.

Managing ACFS Replication

4. Configure the site that is hosting the primary file system.

```
$ /sbin/acfsutil repl init primary \
-s standby_admin/standby_passwd@standby_repl_site \
-m /standby/repl_data -c primary_repl_service \
     /acfsmnts/repl_data
```

5. Monitor information about replication on the file system.
6. Manage the replication background processes.

4. Configure the site that is hosting the primary file system and start ACFS replication by using the `acfsutil repl init primary` command. This command requires the following configuration information:
   - The connect string to be used to connect to the site that is hosting the standby file system, for example: `standby_admin/standby_passwd@standby_repl_site`
     The user `standby_admin` must have `SYSASM` and `SYSDBA` privileges.
   - The mount point of the primary file system, for example: `/acfsmnts/repl_data`
   - If the primary file system is using a different service name than the standby file system, the `–c` option
   - If the mount point is different on the site that is hosting the standby file system than it is on the site that is hosting the primary file system, the mount point specified with the `–m` option
5. Monitor information about replication on the file system. The `acfsutil repl info` command displays information about the state of replication processing on either system.
6. Manage the replication background processes. Run the `acfsutil repl bg` command to start, stop, or retrieve information about the replication background processes.

Run `acfsutil repl pause` to momentarily stop replication, if needed. You should run the `acfsutil repl resume` command soon as possible to resume replication.

Using Replication in Conjunction with ACFS Security and Encryption

With Oracle Database 12*c*, ACFS replication can be used in conjunction with ACFS security and encryption.

- This capability enables:
  - Replication of realm-secured file systems
  - Replication of encrypted file systems
  - Realm security to be configured on an existing replicated file system
  - Encryption to be configured on an existing replicated file system
- The replicated file system inherits the security policies and encryption settings from the primary file system.

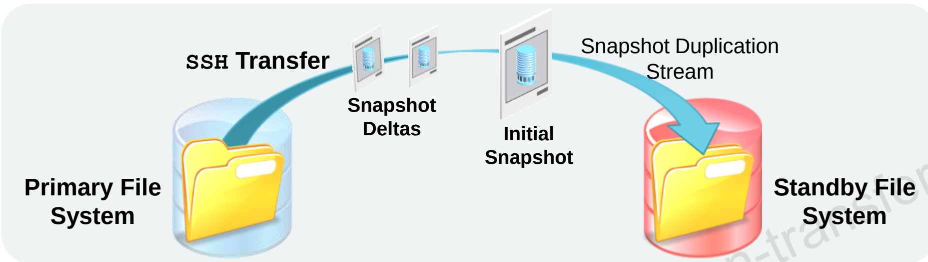
ORACLE

With Oracle Database 12*c*, ACFS replication can be used in conjunction with ACFS security and encryption. This new capability essentially lifts the previous restriction that disallowed replication of secured or encrypted file systems. With this enhancement, file systems that are configured with security or encryption (or both) can be replicated. Likewise, existing replicated file systems can be configured to implement security or encryption (or both).

When replication is used in conjunction with security or encryption, the replicated file system inherits the security policies or encryption settings from the primary file system. Otherwise, the configuration and management of replication, security, and encryption are not altered.

Oracle ACFS Snapshot-Based Replication

- ACFS replication enables replication of Oracle ACFS file systems across a network to a remote site.
- Oracle ACFS snapshot-based replication operates by:
  1. Recording snapshots of the primary file system
  2. Transferring changes between successive snapshots of the primary to the standby file system by using `ssh`
- Replication operations can either occur in constant mode or be scheduled to occur at fixed intervals.



ORACLE

Oracle ACFS snapshot-based replication transfers differences between successive snapshots of the primary file system to the standby file system by using the standard `ssh` protocol.

Oracle ACFS replication enables replication of Oracle ACFS file systems across a network to a remote site, providing disaster recovery capability for the file system. The source Oracle ACFS file system of an Oracle ACFS replication relationship is referred to as a primary file system. The target Oracle ACFS file system of an Oracle ACFS replication relationship is referred to as a standby file system.

Oracle ACFS snapshot-based replication operates by recording snapshots of the primary file system. After the initial snapshot is transferred to the standby file system, replication continues by transferring the changes between successive snapshots of the primary to the standby file system. These replication operations can occur in constant mode (enabling a new operation to start as soon as the previous one completes), or can be scheduled to occur at fixed intervals. This replication solution is by nature asynchronous.

Snapshot Replication Basics

- The source ACFS file system of an ACFS replication relationship is referred to as a *primary file system*.
- The target ACFS file system of an ACFS replication relationship is referred to as a *standby file system*.
- A site can host both primary and standby file systems.
- An ACFS file system cannot be used simultaneously as a primary and a standby file system.

The source Oracle ACFS file system of an Oracle ACFS replication relationship is referred to as a primary file system. The target Oracle ACFS file system of an Oracle ACFS replication relationship is referred to as a standby file system.

A site can host both primary and standby file systems. For example, if there are cluster sites A and B, a primary file system hosted at site A can be replicated to a standby file system at site B. Also, a primary file system hosted at site B can be replicated to a standby file system at site A. However, an Oracle ACFS file system cannot be used simultaneously as a primary and a standby file system.

ACFS Snapshot-Based Replication with ACFSUTIL

- Oracle ACFS snapshot replication enhancements include:
  - `ACFSUTIL` operating system commands to create and manage snapshot clones
  - Additional commands to provide support for duplicating, remastering, and renaming snapshots
- New commands include:
  - `acfsutil snap duplicate create`
  - `acfsutil snap duplicate apply`
  - `acfsutil snap remaster`
  - `acfsutil snap rename`

ORACLE

Oracle ACFS snapshot replication enhancements include `acfsutil` operating system commands to create and manage snapshot clones. Additional commands provide support for duplicating, remastering, and renaming snapshots.

The new commands include:

- `acfsutil snap duplicate create`: Creates a duplicate of an existing snapshot
- `acfsutil snap duplicate apply`: Updates a duplicate snapshot
- `acfsutil snap remaster`: Remasters a file system with a specified snapshot
- `acfsutil snap rename`: Renames a snapshot

ACFS Snapshot Replication Requirements

- Oracle ACFS replication uses the `ssh` utility as the transport between the primary and standby clusters.
- To enable automated use of `ssh`, replication requires two kinds of keys to be configured:
  - On each node in the primary cluster, the `root` user must have a host key stored for each node in the standby cluster.
  - On each node of the standby cluster, a designated unprivileged user, the apply user, must have a public key stored for `root`.
- Note that `ssh` is not provided natively on Windows and will have to be installed and configured to support replication.
- Oracle ACFS replication requires the disk group compatibility attributes for ASM and ADVM to be set to 12.2 or later.

ORACLE

Oracle ACFS replication uses the `ssh` utility as the transport mechanism between the primary and standby clusters. To enable automated use of `ssh`, replication requires two kinds of keys to be configured. These keys must be available on each node where replication is enabled to run.

- On each node in the primary cluster, the `root` user, or the local `SYSTEM` on Windows, must have a host key stored for each node in the standby cluster.
- On each node of the standby cluster, a designated unprivileged user, the `apply` user, must have a public key stored for `root` or the local `SYSTEM` that is authorized to log in as the `apply` user on that node.

Note that `ssh` is not provided natively on Windows and will have to be installed and configured to support replication.

Before using replication on a file system, ensure that you have checked the following:

- There is sufficient network bandwidth to support replication between the primary and standby file systems.
- The configuration of the sites hosting the primary and standby file systems enables the standby file system to keep up with the rate of change on the primary file system.
- Host keys and user keys for `ssh` have been configured as described previously.

Before replicating an Oracle ACFS file system, a replication configuration must be established that identifies information such as the site hosting the primary file system, the site hosting the standby file system, the file system to be replicated, the mount point of the file system, and a list of tags if desired.

The primary and standby sites must share the same user and group configurations, including all UIDs and GIDs in use in the file system. The `apply` user described previously must be configured on each standby node where replication is enabled. This user should be a member of the Oracle ASM administration group and the Oracle DBA group.

To use the Oracle ACFS replication functionality, the disk group compatibility attributes for ASM and ADVM must be set to 12.2 or later for the disk groups that contain the primary and standby file systems.

Note

- The Oracle ACFS replication functionality supports only one standby file system for each primary file system.
- The standby file system is read-only for as long as replication is active on it. Read/write snapshots may be created of the standby if desired.
- A primary site that is running Linux, Solaris, or AIX can replicate to a standby site that is running any of these operating systems. A primary site that is running Windows can replicate only to a standby site that is running Windows.
- The primary and standby sites should be running the same version of the Oracle Grid Infrastructure software. When upgrading the sites, update the standby site first.
- Using replication with database files on Oracle ACFS is not supported.
- Oracle ACFS replication is not supported with Oracle Restart.

Configuring SSH for Use with ACFS Replication

1. Choose an Oracle ACFS replication user.
   – The user should have Oracle ASM administration privileges.
2. Distribute the required SSH keys:
   – A public key for `root` on each node of the primary cluster must be known to *`repluser`* on each node of your standby cluster.
   – If a public key does not exist, generate a public and private key pair on the primary by running the following command as `root`.

```
[root@primary ]# ssh-keygen -t rsa
```

   – Append the `root id_rsa.pub` file to each secondary cluster node's `/home/repluser/.ssh/authorized_keys` file.
   – Generate the host keys for the standby cluster by running `ssh` manually as `root` from each primary node to each standby node.

```
[root@primary01 ]# ssh repluser@standby01 date
[root@primary01 ]# ssh repluser@standby02 date
...
```

ORACLE

ACFS snapshot-based replication uses `ssh` as the transport between the primary and standby clusters, so the user identity under which replication is performed on the standby must be carefully managed. In the replication process, the `root` user on the primary node where replication is running uses `ssh` to log in to the standby node that is involved in the replication. Because it is not advisable for `ssh` to log in as `root` on the standby node, a minimally privileged user should be used. The user chosen should have ASM administration privileges.

A public key for `root` that is defined on each node of your primary cluster must be known to *`repluser`* on each node of your standby cluster. To make the key known, the directory `~`*`repluser`*`/.ssh` must exist on each standby node. If this directory does not exist, create it with access only for *`repluser`*.

If a public key for `root` is defined on a given primary node, it should reside in a `.pub` file, such as `/root/.ssh/id_rsa.pub`. Append the key to the `~repluser/.ssh/authorized_keys` file on each standby node. If a public key does not exist, generate a public and private key pair on the primary by running the following command as `root`:

```
# ssh-keygen -t rsa
```

A host key for each standby node where replication may run must be known on each primary node where replication may run. One way to generate the correct key is to run `ssh` manually as `root` from each primary node to each standby node.

If the correct host key is not known already, a warning displays and you can add the key.

Notice that there are two users involved in the `ssh` connection. Whereas `ssh` on the primary node connects to the standby node as `root`, `ssh` logs in on the standby node as *`repluser`*. Any command run by `ssh` on the standby runs with the privileges of *`repluser`*, not `root`.

Because the primary node connects to the standby node as `root`, the host key for the standby node must be added to the `known_hosts` file of the `root` user, not *`repluser`*. To update the host key:

```
[root@primary /]# ssh repluser@standby date
```

Host Key Configuration for VIPs

- After completing the host key setup, you must perform an additional step if you use VIPs to communicate with your standby cluster.
- You must add the VIP name at the start of each line of the `known_hosts` file that refers to a host in the standby cluster.
- Assume a VIP named `standby12_vip`, and the `known_hosts` file contains the following lines that refer to your standby:

```
standby01,10.242.20.22 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC3pM2YTd4UUiEWEoCKDGgaTgsmPkQToD
...
standby02,10.242.20.23 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDIszcjzNtKN03SY8Kl846skFTVP1HF/y
```

- To enable the use of the VIP, modify the two lines as follows:

```
standby12_vip,standby01,10.242.20.22 ssh-rsa
...
standby12_vip,standby02,10.242.20.23 ssh-rsa
```

ORACLE

After the host key setup for standby nodes is complete on a given primary node, you need to perform an additional step if you use a Virtual IP address (VIP) to communicate with your standby cluster. You must add the VIP name or address at the start of each line of the `known_hosts` file that refers to a host in the standby cluster.

For example, if you use a VIP with the name `standby012_vip`, and your `known_hosts` file contains the following two lines that refer to your standby:

```
standby01,10.242.20.22 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC3pM2YTd4UUiEWEoCKDGgaTgsmPkQToDrdt
...
standby02,10.242.20.23 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDIszcjzNtKN03SY8Kl846skFTVP1HF/ykswbmk
...
```

To enable the use of the VIP, you would modify these two lines to read as shown in the slide.

Configuring the Standby Site

- Create a new standby ACFS file system.
  - It must be sized to hold the files replicated from the primary file system, as well as hold a single replication snapshot.
  - It most be mounted on one node only.
- Run the `acfsutil repl init standby` command on the site that is hosting the standby file system.

```
# acfsutil repl init standby -u repluser /standby/repl_data
```

- After `acfsutil repl init standby` is run, mount the standby file system on all nodes of the standby cluster.

ORACLE

Before replicating an Oracle ACFS file system, configure the site that is hosting the standby file system by performing the following:

- Create a new standby ACFS file system of adequate size to hold the files replicated from the primary file system, as well as a single replication snapshot. For example:

  ```
  /standby/repl_data
  ```

  - Mount the file system on one node only.
  - Run the `acfsutil repl init standby` command on the site that is hosting the standby file system. For example:

    ```
    # /sbin/acfsutil repl init standby -u repluser
    /standby/repl_data
    ```

- After the `acfsutil repl init standby` command has successfully completed, mount the specified file system on all nodes of the standby cluster.

Configuring the Primary Site

Run the `acfsutil repl init primary` command on the site that is hosting the standby file system.

```
$ acfsutil repl init primary -i 2h -s repluser@standby12_vip
-m /standby/repl_data /acfsmounts/repl_data
```

- The `-i` option specifies the minimum time between replication operations.
- The `-m` option specifies a standby mount point if it is different from the primary mount point.

After the standby file system has been set up, configure the site that is hosting the primary file system and start replication. Run the `acfsutil repl init primary` command on the site that is hosting the primary file system. For example:

```
$ acfsutil repl init primary -i 2h -s repluser@standby12_vip -m
/standby/repl_data /acfsmounts/repl_data
```

This command requires the following configuration information:

- A replication interval, given with the option `-i interval` for interval mode or the option `-C` for constant mode replication. If an interval is specified, the value is the minimum amount of time that elapses between replication operations. At the start of each operation, replication takes a new snapshot of the primary and compares it to the previous snapshot, if any. The changes needed to update the standby to match the primary are then sent to the standby. If `-C` is given, a new replication operation is started as soon as the previous one completes.
- The username and network endpoint (VIP name or address, or host name or address) to be used to connect to the site that is hosting the standby file system, specified with the `-s` option—for example: `-s repluser@standby12_vip`
- The mount point of the primary file system. For example: `/acfsmounts/repl_data`
- If the mount point is different on the site that is hosting the standby file system, a mount point is specified for the standby file system with the `-m standby_mount_point` option.

Snapshot-Based Replication Upgrade Prerequisites

- Flex ASM must be enabled for any existing instance when the instance is upgraded to Oracle Grid Infrastructure 12.2.
- Before starting the upgrade process, you must configure `ssh` as described previously.
- Before upgrading to snapshot-based replication, upgrade your primary and standby clusters to Grid Infrastructure 12.2.
  - The cluster upgrade should be completed within 24 hours because the clusters continue to run the existing replication.
- Set the `COMPATIBLE.ADVM` disk group attribute of the file systems to be involved in the replication to 12.2.0.0.0.

ORACLE

Oracle Flex ASM must be enabled for any existing instance when the instance is upgraded to Oracle Grid Infrastructure 12*c* release 2. If a password file exists in the instance to support replication, the file must be enabled for use in the new Oracle Flex ASM instance. Snapshot-based replication uses `ssh` as the transport between the primary and standby clusters. Before starting the upgrade process, you must configure `ssh` as described in the previous slides.

Before upgrading to snapshot-based replication, you must upgrade your standby cluster system to Oracle Grid Infrastructure 12*c* release 2 (12.2). This cluster upgrade should be completed within 24 hours. The standby cluster continues to run the existing replication during the upgrade process. Next, upgrade to Oracle Grid Infrastructure 12*c* release 2 (12.2) on the primary cluster. This upgrade should also be completed within 24 hours. During this upgrade, the primary cluster continues to run the existing replication that was installed before 12.2. After the completion of the Oracle Grid Infrastructure 12*c* release 2 (12.2) upgrade of the standby and primary clusters, the existing replication implementation that was installed before 12.2 is not supported.

Immediately following the upgrade of the two clusters, set the `COMPATIBLE.ADVM` disk group attribute associated with the file systems to be involved in the replication to 12.2.0.0.0. For a detailed list of steps required to complete the replication upgrade, refer to *Oracle Automatic Storage Management Administrator's Guide 12c Release 2*.

Snapshot Replication Considerations

- On the standby, a backup snapshot is created at the end of each replication operation.
- This snapshot records the latest consistent contents of the standby.
- The backup snapshot can be used to recover the contents if a permanent outage occurs during the replication operation.
- Each backup snapshot is deleted when the following replication operation completes:
  - So it must always be possible to create a backup snapshot.
- You should ensure that the primary and standby file systems do not run out of disk space.
  - Configure automatic resize to avoid running out of space, if needed.

ORACLE

On the standby, a backup snapshot is created at the end of each replication operation. This snapshot records the latest consistent content of the standby, and can be used to recover the content if a permanent outage occurs during the current replication operation. Each backup snapshot is deleted when the following replication operation is complete, so it must always be possible to create a backup snapshot. In addition, enough space must exist for the version of the standby captured in the snapshot and the current file system content.

You should ensure that the primary and standby file systems do not run out of disk space. If either file system runs out of available storage, you should expand the file system or, on the primary file system or in the read/write snapshots on the standby, remove files from the file system to free up space. You can also configure automatic resize to avoid running out of space.

If the primary file system runs out of space and you decide to free up space by removing files, you should remove only those files that are not being replicated. Replicated files are stored in a snapshot pending transfer to the standby file system and are not deleted. You can delete any Oracle ACFS snapshots that are not created by replication.

Quiz

Which statements are true about ASM Filter Driver (ASMFD)?

a. ASMFD protects database files by rejecting non-Oracle I/O requests.

b. ASMFD drastically reduces the number of open file descriptors that are required to support large numbers of Oracle Database processes.

c. ASMFD allows Oracle Clusterware to perform node-level fencing without a restart.

ORACLE

Answer: a, b, c

Summary

In this lesson, you should have learned how to:

- Configure High Availability NFS
- Describe how to configure ASM Filter Driver
- Upgrade an existing database to use ASM Filter Driver and see its effect
- Configure and manage ACFS auditing
- Implement ACFS encryption
- Configure and manage ACFS replication
- Configuring and manage ACFS Snapshot-Based replication

B

Oracle Database Exadata Cloud Service Overview

ORACLE

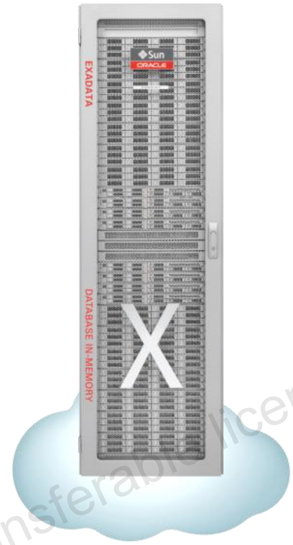
Objectives

After completing this lesson, you should be able to:

- Describe the architecture and capabilities of Exadata Cloud Service
- Compare and contrast between Exadata Cloud Service and an on-premise Exadata implementation

Introducing Exadata Cloud Service

- Oracle Database with all features and options:
  – Industry-leading database for mission-critical OLTP and analytics
- On Exadata Database Machine:
  – The fastest and most available database cloud platform
- In the Oracle Cloud:
  – No capital expenditure, just a simple monthly subscription
  – Oracle deploys and manages the infrastructure
  – Fast, elastic, web-driven service provisioning
  – Complete service isolation with no over-provisioning
  – 100% compatibility with on-premises applications and Oracle database



ORACLE

Oracle Database Exadata Cloud Service enables you to leverage the power of Oracle Exadata Database Machine inside the Oracle Cloud. With Exadata Cloud Service you get:

- Oracle Database with all features and options:

Each Exadata Cloud Service database deployment is provisioned with a complete Oracle Database installation that includes all the features of Oracle Database Enterprise Edition, plus all the database enterprise management packs and all the Enterprise Edition options, such as Oracle Real Application Clusters (RAC), Oracle Database In-Memory, and Oracle Multitenant.

- On Exadata Database Machine:

The foundation for Exadata Cloud Service is Exadata Database Machine. Deployed at thousands of sites around the world, Exadata is established as the highest performing and most available platform for running Oracle Database. With a fault-tolerant architecture featuring scale-out database servers and scale-out intelligent storage connected with a fully redundant high-performance InfiniBand fabric, Exadata is an ideal cloud platform.

Exadata Cloud Service delivers all of the advanced features of Exadata, including SQL offload, Smart Flash Cache, Storage Index, and Hybrid Columnar Compression.

- In the Oracle Cloud:

Customers pay a simple monthly subscription fee for Exadata Cloud Service. There is no initial capital cost and no data center costs.

All supporting infrastructure for Exadata Cloud Service is deployed, maintained and managed by Oracle, including datacenter networking, private Exadata InfiniBand networks, physical Exadata database servers and storage servers, firmware, and Exadata Storage Server software.

Exadata Cloud Service includes easy-to-use web-based wizards through which you can quickly provision an Exadata system and associated database deployments. The wizards are available through Oracle's Cloud Portal, http://cloud.oracle.com.

Exadata Cloud Service features complete service isolation with no over-provisioning of hardware to ensure that response times and throughput are predictable for critical business processes. This contrasts with some cloud service delivery models that silently overprovision hardware, and consequently may not be able to deliver the expected resources during busy periods.

Exadata Service is 100% compatible with on-premises Oracle databases and all existing applications. Exadata Cloud Service enables existing on-premises Exadata customers to easily embark on a journey to the cloud – without compromising the database performance and availability levels they enjoy with their on-premises Exadata deployments. While existing Oracle Database customers who have not yet experienced Exadata can easily start enjoying the performance, availability and scalability benefits of Exadata without compromising any of the database functionality that they rely on. With Exadata Cloud Service, organizations can easily deploy a hybrid cloud environment that uses on-premises databases as well as databases in the Cloud.

Service Configuration Options

	Quarter Rack	Half Rack	Full Rack
Number of Database Servers	2	4	8
Number of CPU Cores [1]	16 - 68	56 - 136	112 – 272
Total RAM Capacity	496 GB	992 GB	1984 GB
Number of Exadata Storage Servers [2]	3	6	12
Total Flash Capacity	19.2 TB	38.4 TB	76.8 TB
Total Usable Disk Capacity [3]	42 TB	84 TB	168 TB
Maximum SQL Flash Bandwidth [4]	30 GB/sec	60GB/sec	120GB/sec
Maximum SQL Flash Read IOPS [5]	900,000	1,800,000	3,600,000
Maximum SQL Flash Write IOPS [6]	500,000	1,000,000	2,000,000
Maximum SQL Disk Bandwidth [4]	4.5 GB/sec	9 GB/sec	20 GB/sec
Maximum SQL Disk IOPS [5]	7,000	14,000	28,000
Maximum Data Load Rate [7]	5 TB/hour	10 TB/hour	10 TB/hour

ORACLE

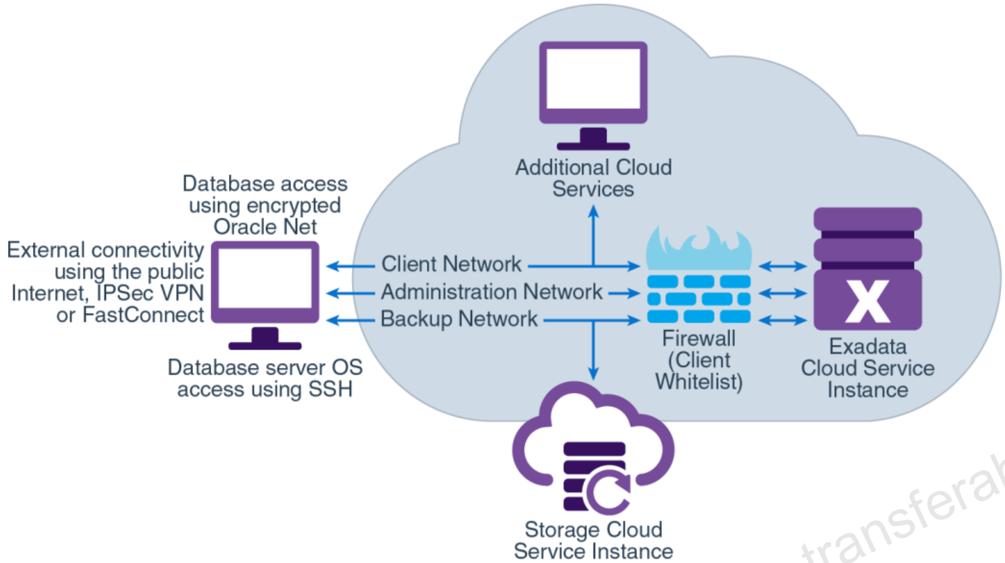
Each Exadata Cloud Service instance is based on an Exadata configuration that contains a predefined number of database servers and a predefined number of Exadata Storage Servers, all tied together by a high-speed, low-latency InfiniBand network and intelligent Exadata software. Three configurations are offered; Quarter Rack, Half Rack and Full Rack. The table in the slide outlines the vital statistics for each system configuration.

Note the following:

1. Each Exadata Cloud Service configuration is equipped with a fixed amount of memory, storage and network resources. However, you can choose how many database server CPU cores are enabled, within the minimum and maximum limits listed in the table. This enables you to scale an Exadata Cloud Service configuration to meet workload demands, and only pay for the database server resources that you need. Each database server must contain the same number of enabled CPU cores.
2. Note that Exadata Cloud Service Half Rack and Full Rack configurations differ from on-premises Half Rack and Full Rack. The Exadata Cloud Service Half Rack contains six storage servers, and the Exadata Cloud Service Full Rack contains twelve storage servers. Consequently, an Exadata Cloud Service Half Rack is twice the size of a Quarter Rack, and a Full Rack is twice the size of a Half Rack.
3. The usable storage capacity is the storage that available for Oracle Database files after taking into account high-redundancy ASM mirroring (triple mirroring), which is used to provide highly resilient database storage on all Exadata Cloud Service configurations. The usable storage capacity does not factor in the effects of Exadata compression capabilities, which can be used to increase the effective storage capacity.
4. Bandwidth is the peak physical scan bandwidth that is achieved by running SQL, and assuming that there is no database compression. Effective user data bandwidth is higher when database compression is used. In all cases, actual performance varies by application.

5. Based on 8K Oracle Database I/O requests. Note that the I/O size greatly affects Flash IOPS, so IOPS based on smaller I/Os is not relevant for databases.
6. Based on 8K Oracle Database I/O requests. Flash write I/Os are measured at the storage servers after ASM mirroring, which issues multiple storage I/Os to maintain redundancy.
7. Load rates are typically limited by database server CPU capacity, not I/O. Rates vary based on a variety of factors, including the load method and data types used, and the use of indexes, compression and partitioning.

Service Connection Options



ORACLE

The diagram in the slide outlines the primary ways of connecting to Exadata Cloud Service.

You can connect directly from cloud clients, which are applications already running in the Oracle Cloud. This includes java applications connecting through JDBC running on the Java Cloud Service, or client applications connecting through Oracle Net (SQL*Net) running on a Compute Cloud Service instance.

You can also connect to Exadata Cloud Service from your on-premises applications by using Oracle Net. On Exadata Cloud Service, Oracle Net is configured by default to use native encryption and integrity capabilities to secure data in transit.

Access to the database server operating system is provided using Secure Shell (SSH). This is primarily used for administration purposes.

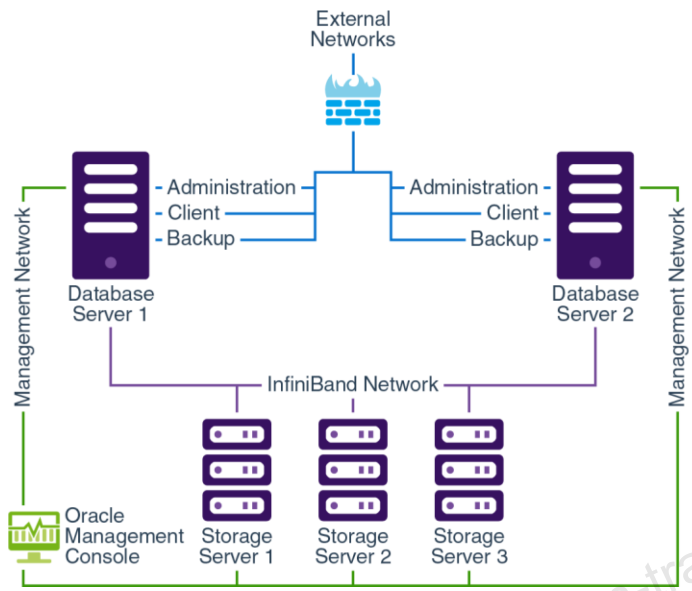
A backup network is also provided. This network keeps high-load activities separate from application connections and is primarily used when Exadata Cloud Service database deployments are backed up to an Oracle Storage Cloud Service container.

The Oracle Cloud network is firewall-protected and you must specify the clients that can access the service. The permitted clients are registered in the firewall using a whitelist.

To provide connectivity between your network and the Oracle Cloud you can:

- Use the public Internet and secure protocols, such as SSH and encrypted Oracle Net.
- Configure an IPSec VPN to enable secure connectivity between your network and the Oracle Public Cloud over the Internet.
- Use Oracle Network Cloud Service - FastConnect Partner Edition, which is a secure, high-bandwidth, point-to-point networking solution that is integrated with Oracle Cloud.

Service Architecture



The diagram in the slide illustrates the Exadata Cloud Service architecture for a Quarter Rack service instance. Larger service instances are principally the same, except that they contain more database servers and Exadata Storage Servers.

The architecture is essentially the same as for an on-premises implementation of Exadata, with clustered database servers connected to a grid of Exadata Storage Servers through a high-speed, low-latency, InfiniBand network.

Application users and administrators can connect only to the database servers, using the supplied administration, client and backup network interfaces.

Oracle manages all hardware, firmware, and the Exadata Storage Server software by using a separate management network.

Service Availability

- Exadata Cloud Service inherent HA capabilities:
  - Exadata platform full hardware redundancy:
    - Database and storage servers
    - InfiniBand and Ethernet networking
    - Power supplies and PDUs
  - Oracle software HA capabilities:
    - RAC protects against database server failures
    - ASM data mirroring protects against storage failures
      - Exadata Cloud Service used 3-way mirroring (ASM high redundancy)
    - Plus Flashback technologies, Online DDL, In-memory fault-tolerance, and so on...
- Implementation best-practices are derived from thousands of mission-critical deployments worldwide

Exadata Cloud Service inherits all of the high availability (HA) capabilities that are an integral part of the Exadata platform. Exadata has full hardware redundancy including redundant InfiniBand networking, redundant network ports, redundant Power Distribution Units (PDUs), redundant power supplies, redundant database servers, and redundant storage servers.

Exadata Cloud Service inherits all of the Oracle Database HA capabilities. Oracle RAC protects against database server failures. Automatic Storage Management (ASM) provides data mirroring to protect against disk or storage server failures. In Exadata Cloud Service, high redundancy (3-way mirroring) is used by default to protect data. Oracle Recovery Manager (RMAN) provides extremely fast and efficient backups to disk or cloud storage. Oracle Database Flashback technologies allow user errors to be backed out at the database, table or row level. Online Data Definition Language (DDL) operations enhance availability during data maintenance operations. In-memory fault-tolerance mirrors the contents of the in-memory data store, which enables surviving stores to automatically and transparently satisfy user queries if a database instance fails.

Because of its HA pedigree, Exadata is deployed worldwide at thousands of mission-critical deployments in leading banks, airlines, telecommunications companies, stock exchanges, government agencies and e-commerce sites. The best practices guiding the implementation of Exadata Cloud Service are derived from the experience of those deployments.

Management Responsibilities

Oracle Managed (No Customer Access)	Customer Managed (No Oracle Access)
Initial configuration and installation	Database server DomU - including OS
Exadata Storage Server software and objects	Oracle Database, Grid Infrastructure, ASM
Database server Dom0	Database and OS updates*
InfiniBand switches, HCAs, and partitioning	Database and OS monitoring*
Management and ILOM networks	Database backup and recovery*
All hardware, firmware, and BIOS	
Client access VLANS and IP addresses	* Oracle tooling provided

The table in the slide summarizes the division of management responsibilities for Exadata Cloud Service. Additional details are provides in the following notes and throughout this lesson.

Oracle manages the following elements, which customers cannot access:

- Initial configuration and installation of all hardware and software components.
- Exadata Storage Server software and all Exadata Storage Server objects (cell disks, grid disks, flash cache and so on).
- Database server Dom0 environment, including the operating system (OS) and hypervisor.
- InfiniBand switches, Host Channel Adapters (HCAs), and InfiniBand partitioning.
- Dom0 management network and Integrated Lights Out Manager (ILOM) network.
- All hardware, firmware and BIOS.
- IP addresses and VLAN configuration for the client and backup networks.

Customers manage the follow elements, which Oracle cannot access:

- Database server virtual machine (DomU), including the kernel and other OS components.
- Oracle Database, Grid Infrastructure, and Automatic Storage Management (ASM). Customers manage all aspects of their Oracle databaseroless, including creation and maintenance of tablespaces, tables, indexes, users,, and so on.
- Patching and updates for Oracle Database software and DomU OS components. Oracle provides Exadata Cloud Service-specific tools to assist.
- Oracle Database and OS monitoring. Exadata Cloud Service is configured with Enterprise Manager Database Control 11*g* and Enterprise Manager Express 12*c* for local database management. Customers may manage their Exadata Cloud Service environment using a separate Oracle Enterprise Manager Cloud Control environment.
- Database backup and recovery. Oracle provides Exadata Cloud Service-specific tools to assist. Also, during database configuration you can choose to configure automated daily database backups.

Storage Configuration

- Preconfigured ASM Disk Groups:
  - High redundancy disk groups, which consume nearly all of the storage
    - DATA contains Oracle Database data files
    - RECO contains the Fast Recovery Area (FRA)
  - System disk groups, which are comparatively very small
    - DBFS contains shared clusterware files
    - ACFS disk groups contain Oracle Database binaries and patch files
- Configuration options for space allocation:
  - Provision for Local Backups: 40% DATA, 60% RECO
  - No Provision for Local Backups: 80% DATA, 20% RECO

	Quarter Rack	Half Rack	Full Rack
Total Usable Disk Capacity	42 TB	84 TB	168 TB
Disk Group Sizes ***with*** Provision for Local Backups	DATA: 16.8 TB RECO: 25.2 TB	DATA: 33.6 TB RECO: 50.4 TB	DATA: 67.2 TB RECO: 100.8 TB
Disk Group Sizes ***without*** Provision for Local Backups	DATA: 33.6 TB RECO: 8.4 TB	DATA: 67.2 TB RECO: 16.8 TB	DATA: 134.4 TB RECO: 33.6 TB

ORACLE

As part of provisioning each Oracle Exadata Database Machine environment, the storage space inside the Exadata Storage Servers is provisioned for use by Oracle Automatic Storage Management (ASM). By default, the following ASM disk groups are created:

- Two high redundancy disk groups consume nearly all of the available storage space:
  - The DATA disk group is intended for the storage of Oracle Database data files.
  - The RECO disk group is primarily used for storing the Fast Recovery Area (FRA), which is an area of storage where Oracle Database can create and manage various files related to backup and recovery, such as RMAN backups and archived redo log files.
- The system disk groups support various operational purposes.
  - The DBFS disk group is primarily used to store the shared clusterware files (Oracle Cluster Registry and voting disks).
  - The ACFS disk groups are primarily used to store Oracle Database binaries and to facilitate patching.

  Compared to the DATA and RECO disk groups, the system disk groups are very small, consuming less than 1% of the total available disk space. The system disk groups are configured to use normal redundancy. You should not store Oracle Database data files or backups inside the system disk groups.

Note that the disk group names contain a short identifier string that is associated with your Exadata Database Machine environment. For example, the identifier could be `C2`, in which case the DATA disk group would be named `DATAC2`, the RECO disk group would be named `RECOC2`, and so on.

As an input to the provisioning process, you must decide if you intend to perform backups to the local storage within your Exadata Database Machine. Your backup storage choice profoundly affects how storage space in the Exadata Storage Servers is allocated to the ASM disk groups.

If you choose to provision for local backups, approximately 40% of the available storage space is allocated to the DATA disk group and approximately 60% is allocated to the RECO disk group. If you choose not to provision for local backups, approximately 80% of the available storage space is allocated to the DATA disk group and approximately 20% is allocated to the RECO disk group.

After the Exadata Database Machine is activated, the only way to adjust the storage allocation is by lodging a Service Request with Oracle. For details see My Oracle Support Note 2007530.1.

The table in the slide outlines how the usable storage capacity is allocated to the DATA and RECO disk groups for each configuration option. The usable storage capacity is the storage that available for Oracle Database files after taking into account high-redundancy ASM mirroring (triple mirroring), which is used to provide highly resilient database storage on all Exadata Cloud Service configurations. The usable storage capacity does not factor in the effects of Exadata compression capabilities, which can be used to increase the effective storage capacity. The usable disk capacity does not include the space allocated to the system disk groups.

Storage Management Details

- Storage is preconfigured and allocated to ASM disk groups
- Customers manage database objects inside ASM
- Oracle manages the Exadata storage, including Exadata Storage Server software updates
- Exadata Storage Servers are configured using best practices:
  - One cell disk on each physical storage device
  - One set of grid disks for each ASM disk group
    - Space allocation depends on backup configuration
    - Disk groups for database files use high redundancy
    - Grid disk names are prefixed with the corresponding disk group name
  - Flash cache and flash log are preconfigured
    - Flash cache mode is write-through
  - IORM plan is active, IORM objective is set to balanced
- Customers have no direct access to the Exadata Storage Servers
  - Can request custom configurations for grid disks and IORM

ORACLE

With Exadata Cloud Service, storage is preconfigured and allocated to ASM disk group, so customers must manage database objects inside ASM. This includes management of tablespaces and the data they contain. Customers must also monitor and maintain ASM to ensure continued availability.

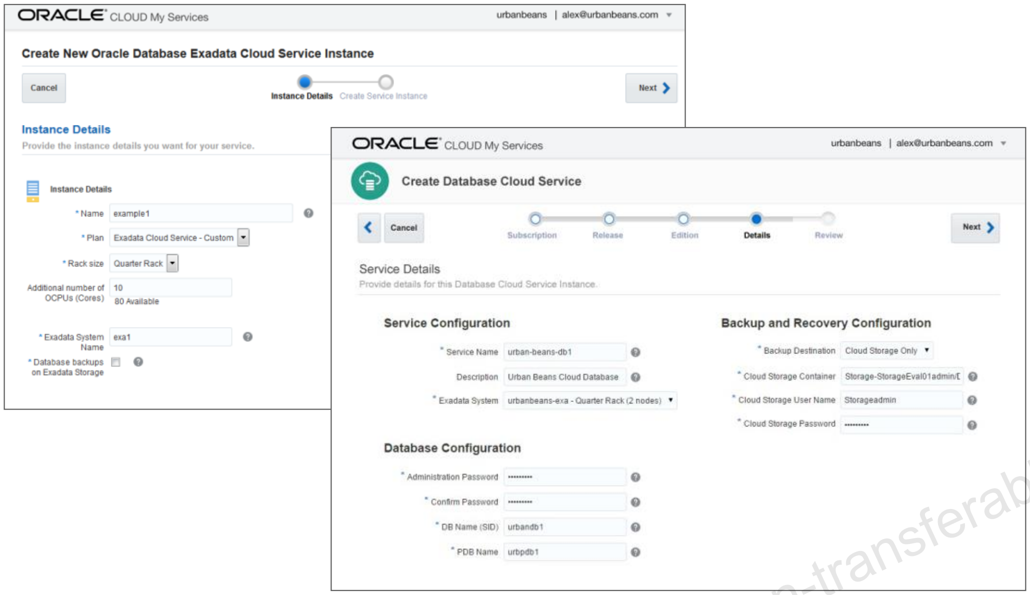
Oracle manages the Exadata Storage Servers, including Exadata Storage Server software updates. The Exadata Storage Servers are configured in accordance with best practices:

- The cell disks, and grid disks to support the DATA, RECO and system disk groups are preconfigured and use standard Exadata default settings.
- The flash cache and flash log are preconfigured. For Exadata Cloud Service, the flash cache is configured in write-back mode by default.
- The I/O Resource Management (IORM) plan is active and the objective is set to balanced.

No direct access is provided to the Exadata Storage Servers. However, you can lodge a Service Request (SR) for:

- Storage reconfiguration to redistribute the space amongst the existing grid disks, or to create additional grid disks.
- IORM configuration to a custom specification.

Simple Web-Based Provisioning

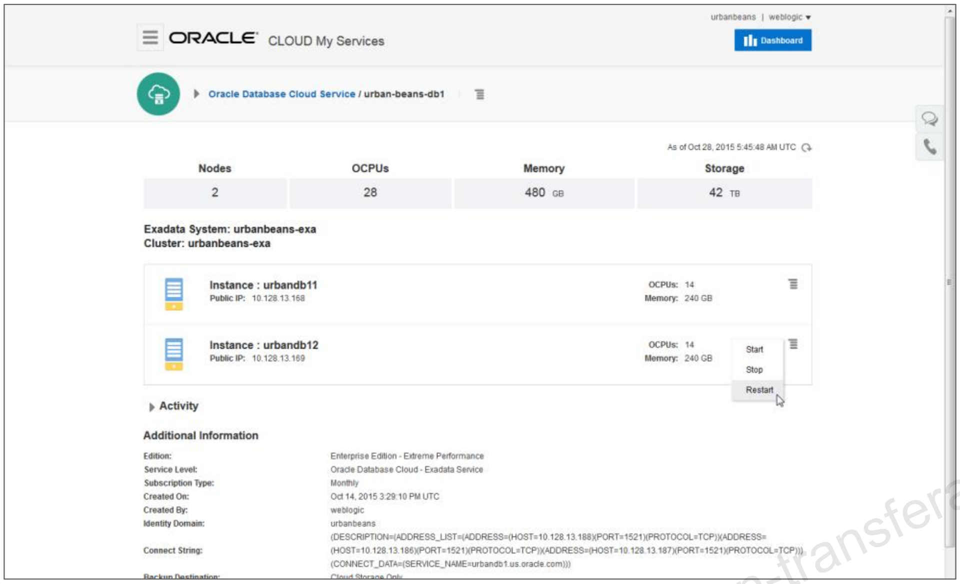


ORACLE

Exadata Cloud Service provides simple web-based provisioning interfaces, which you can use to quickly and easily provision an Exadata system and subsequently create their Oracle databases.

The screenshots in the slide show examples of the key screens in the Exadata Cloud Service instance provisioning wizard, and the wizard used to create databases on Exadata Cloud Service. In both cases, only a few key inputs are required and either task can be completed in only a few minutes.

Simple Web-Based Management



Exadata Cloud Service also provides a set of web-based management interfaces, which you can use to quickly and easily accomplish various management tasks including:

- Database server VM lifecycle activities (start, stop, restart).
- SSH key management (add, update).
- View service instance details.
- View database deployment details.
- Configure an Exadata I/O Resource Management (IORM) inter-database plan.
- Scale within an Exadata system (add or remove database server CPU cores).

The screenshot in the slide show a simple example of the interface that displays database deployment details along with the options to start, stop, or restart one of the database server virtual machines.

REST APIs

- REST APIs provide programmatic management and control:
  - Create or Delete a Database Deployment
  - Stop, Start or Restart a Database Server VM
  - View Details
    - View a Database Deployment
    - View All Database Deployments
    - View Database Servers
    - View the Status of an Operation

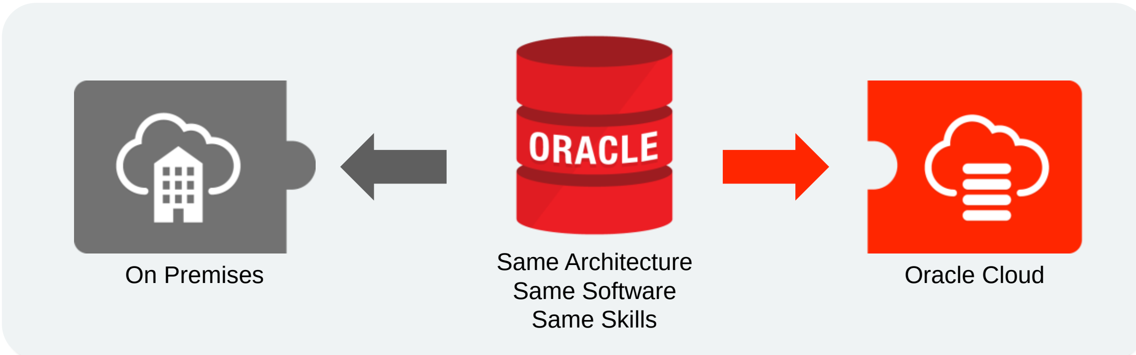
```
$ curl --include --request POST --cacert ~/cacert.pem
  --user serviceadmin:Pa55_word
  --header "X-ID-TENANT-NAME:usexample"
  --header "Content-Type:application/json"
  --data '{ "lifecycleState" : "stop", "vmName" : "node02" }'
  https://dbaas.oraclecloud.com/paas/service/dbcs/api/v1.1/instances/usexample/db12c
```

ORACLE

In addition to the web-management interfaces, Exadata Cloud Service provides a series of REST APIs. The Exadata Cloud Service REST APIs enable you to perform various management operations programmatically, such as create and delete database instances, start and stop compute nodes, and view status information. The slide shows an example that uses the cURL utility to invoke the REST API call to stop a database server virtual machine.

See http://docs.oracle.com/cloud/latest/exadatacs/EXARS/index.html for details about the Exadata Cloud Service REST APIs.

Migrating to Exadata Cloud Service



- Logical Migration options include:
  - Oracle Data Pump
  - Oracle GoldenGate
- Physical Migration options include:
  - Recovery from an RMAN backup
  - Transportable Tablespaces
  - Oracle Data Guard

Full compatibility between existing on-premises Oracle databases and Oracle databases on Exadata Cloud Service makes migration easy and low risk. Following established best practices for Oracle Database, two types of migration methodologies are supported:

- Logical Migration allows data reorganization as part of migration. Database solutions that can be used for this purpose include Oracle Data Pump and Oracle GoldenGate.
- Physical Migration, which involves a byte-to-byte copy of the data, offers the simplest way to migrate databases. Solutions that can be used for this purpose include RMAN backup, Transportable Tablespaces, and Oracle Data Guard. You can also restore from a backup stored on the Oracle Public Cloud through the Oracle Database Backup Service.

Summary

In this lesson, you should have learned how to:

- Describe the architecture and capabilities of Exadata Cloud Service
- Compare and contrast between Exadata Cloud Service and an on-premise Exadata implementation