ORACLE UNIVERSITY

Hardware and Software
Engineered to Work Together

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

Student Guide
D81242GC10
Edition 1.0 | September 2014 | D85442



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

ORACLE

**Author**
Jim Womack

**Technical Contributors and Reviewers**
Harald Van Breederode
Joel Goodman
Jim Williams
Allan Graves
Gerlinde Frenzen
Harald Van Breederode
Joel Goodman
Jim Williams
Ranbir Singh
Andy Fortunak
Al Flournoy
Markus Michalewicz

**Editors**
Arijit Ghosh
Anwesha Ray

**Graphic Designer**
Maheshwari Krishnamurthy

**Publishers**
Joseph Fernandez
Michael Sebastian Almeida
Sumesh Koshy

# Contents

**1 ASM Overview**

Objectives 1-2
What Is Oracle ASM? 1-3
ASM CloudFS and ACFS 1-4
Oracle Flex ASM 1-6
ASM Features and Benefits 1-7
ASM Instance Designs: Nonclustered ASM and Oracle Databases 1-8
ASM Instance Designs: Clustered ASM for Clustered Databases 1-9
ASM Instance Designs: Clustered ASM for Mixed Databases 1-10
ASM Components: Software 1-11
ASM Components: ASM Instance 1-12
ASM Components: ASM Instance Primary Processes 1-14
ASM Components: Node Listener 1-15
ASM Components: Configuration Files 1-16
ASM Components: Group Services 1-17
ASM Components: ASM Disk Group 1-18
ASM Disk Group: Failure Groups 1-19
ASM Components: ASM Disks 1-20
ASM Components: ASM Files 1-21
ASM Files: Extents and Striping 1-22
ASM Files: Mirroring 1-23
ASM Components: ASM Clients 1-25
ASM Components: ASM Utilities 1-26
ASM System Privileges 1-27
ASM OS Groups with Role Separation 1-28
Authentication for Accessing ASM Instances 1-29
ASMCMD and Authentication 1-30
Password-Based Authentication for ASM 1-32
Using a Single OS Group 1-33
Using Separate OS Groups 1-34
ASM Scalability 1-35
Quiz 1-37
Summary 1-38

**2 Administering ASM Instances**
Objectives 2-2
Managing ASM with ASMCA 2-3
Starting and Stopping ASM Instances Using ASMCA 2-4
Starting and Stopping ASM Instances Using ASMCMD 2-5
Starting and Stopping ASM Instances Using srvctl 2-6
Starting and Stopping ASM Instances by Using SQL*Plus 2-7
Starting and Stopping ASM Instances Containing Cluster Files 2-9
ASM Initialization Parameters 2-10
ASM_DISKGROUPS 2-11
Disk Groups Mounted at Startup 2-12
ASM_DISKSTRING 2-13
ASM_POWER_LIMIT 2-15
INSTANCE_TYPE 2-16
MEMORY_TARGET 2-17
Adjusting ASM Instance Parameters in SPFILEs 2-18
Managing the ASM Password File 2-19
Managing a Shared Password File in a Disk Group 2-20
Starting and Stopping the Node Listener 2-22
ASM Dynamic Performance Views 2-23
ASM Dynamic Performance Views Diagram 2-24
Quiz 2-26
Summary 2-28
Practice 2 Overview: Administering ASM Instances 2-29

**3 Flex ASM**
Objectives 3-2
Flex ASM: Overview 3-3
Flex ASM and Flex Clusters 3-4
ASM Instance Changes 3-5
ASM Network 3-6
ASM Listeners 3-7
ADVM Proxy 3-8
Configuring Flex ASM on a Standard Cluster 3-9
Configuring Flex ASM on a Flex Cluster 3-10
Managing Flex ASM Instances 3-11
Stopping, Starting, and Relocating Flex ASM Instances 3-12
Setting the Cardinality for Flex ASM Instances 3-13
Monitoring Flex ASM Connections 3-14
Relocating an ASM Client 3-15
Flex ASM Deployment: Example 3-16

Quiz 3-18
Summary 3-21
Practice 3 Overview: Database Fail Over with Flex ASM 3-22

**4 Administering ASM Diskgroups**
Objectives 4-2
Disk Group: Overview 4-3
Creating a New Disk Group with ASMCMD 4-5
Creating an ASM Disk Group with ASMCA 4-6
Creating an ASM Disk Group: Advanced Options 4-7
Creating a Disk Group with Enterprise Manager 4-8
Creating a Disk Group with SQL*Plus 4-9
Specifying Content Type for a Disk Group 4-11
Renaming Disks Groups 4-13
Disk Group Attributes 4-15
Viewing Disk Group Attributes 4-19
Compatibility Attributes 4-20
Features Enabled by Disk Group Compatibility Attributes 4-21
Support for 4 KB Sector Disk Drives 4-22
Supporting 4 KB Sector Disks 4-23
ASM Support for 4 KB Sector Disks 4-24
Sector Size Validations 4-25
Using the SECTOR_SIZE Clause 4-26
Viewing ASM Disk Groups 4-27
Viewing ASM Disk Information 4-29
Extending an Existing Disk Group 4-31
Dropping Disks from an Existing Disk Group 4-32
REBALANCE POWER 0 4-33
V$ASM_OPERATION 4-34
Adding and Dropping in the Same Command 4-36
Undropping Disks in Disk Groups 4-37
Replacing Disks in Disk Groups 4-38
Renaming Disks in Disk Groups 4-39
Resizing Disks in Disk Groups 4-40
Mounting and Dismounting Disk Groups 4-41
Viewing Connected Clients 4-42
Dropping Disk Groups 4-43
ASM Disk Group Rebalance: Review 4-44
Priority Ordered Rebalance 4-45
Tuning Rebalance Operations 4-46
Proactively Validating Data Integrity 4-48

Proactive Content Checking During Rebalance 4-49
On-Demand Scrubbing 4-50
Errors During Scrubbing 4-51
Checking the Consistency of Disk Group Metadata 4-52
Managing Capacity in Disk Groups 4-53
ASM Fast Mirror Resync: Review 4-56
Controlling the Resources Used by Resync 4-57
Resync Checkpoint and Auto-Restart 4-58
Resync Time Estimate 4-59
Even Read 4-60
Preferred Read Failure Groups 4-61
Dealing with Transient Failure on a Failure Group 4-62
Preferred Read Failure Groups: Best Practice 4-63
Viewing ASM Disk Statistics 4-64
Performance, Scalability, and Manageability Considerations for Disk Groups 4-66
Quiz 4-67
Summary 4-69
Practice 4 Overview: Administering ASM Disk Groups 4-70

**5 Administering ASM Files, Directories, and Templates**
Objectives 5-2
ASM Clients 5-3
Interaction Between Database Instances and ASM 5-5
Accessing ASM Files by Using RMAN 5-6
Accessing ASM Files by Using XML DB 5-8
Accessing ASM Files by Using DBMS_FILE_TRANSFER 5-9
Accessing ASM Files by Using ASMCMD 5-10
Fully Qualified ASM File Names 5-11
Other ASM File Names 5-13
Valid Contexts for the ASM File Name Forms 5-15
Single File Creation: Examples 5-16
Multiple File Creation: Example 5-17
View ASM Aliases, Files, and Directories 5-18
Viewing ASM Files 5-20
ASM Directories 5-21
Managing ASM Directories 5-22
Managing Alias File Names 5-23
Disk Group Templates 5-24
Viewing Templates 5-26
Managing Disk Group Templates 5-27
Managing Disk Group Templates with ASMCMD 5-28

Using Disk Group Templates 5-29
Intelligent Data Placement 5-30
Enabling Intelligent Data Placement 5-31
Viewing Disk Region Information 5-32
Assigning Files to Disk Regions with Enterprise Manager 5-33
ASM Access Control Lists 5-34
ASM File Access Control Available on Windows 5-35
ASM ACL Prerequisites 5-36
Managing ASM ACL with SQL Commands 5-37
Managing ASM ACL with ASMCMD Commands 5-38
Managing ASM ACL with Enterprise Manager 5-39
ASM ACL Guidelines 5-40
Quiz 5-41
Summary 5-43
Practice 5 Overview: Administering ASM Files, Directories, and Templates 5-44

**6 Administering Oracle CloudFS**
Objectives 6-2
Oracle Cloud File System 6-3
Enhanced Platform Support for Cloud FS Data Services 6-4
ASM Files and Volumes 6-5
ACFS and ADVM Architecture: Overview 6-6
ADVM and ACFS Processes 6-8
Oracle ACFS 6-9
Space Allocation 6-10
Striping Inside the Volume 6-11
Volume Striping: Example 6-12
Creating an ACFS Volume 6-14
Managing Dynamic Volumes with SQL*PLUS 6-15
Creating an ACFS Volume with ASMCA 6-16
Creating the ACFS File System with ASMCA 6-17
Register and Mount the ACFS File System with ASMCA 6-18
Using ASMCMD for Dynamic Volumes 6-19
Linux-UNIX Extensions 6-20
ACFS Utilities for Multiple Environments 6-21
Configuration Settings for Database Files 6-22
Cloud FS Cluster Resources 6-23
Implementing Node-Specific File System Dependencies 6-25
ACFS Snapshots 6-26
Managing ACFS Snapshots 6-27
Snapshot Enhancements 6-29

ACFS Backups 6-30
ACFS Performance 6-31
ACFS Views 6-32
Quiz 6-33
Summary 6-34
Practice 6 Overview: Administering Oracle CloudFS 6-35

**7 Oracle CloudFS Advanced Topics**
Objectives 7-2
Cloud FS Auditing 7-3
Cloud FS Audit Trail Files 7-4
Audit Trail Contents 7-5
Configuring Cloud FS Auditing 7-6
Initializing Auditing Roles and Enabling Audit Sources 7-7
Enabling Auditing of Command Rules in a Security Realm 7-8
Managing the Audit Trail 7-9
Archiving Audit Files 7-10
Reviewing Audit Files 7-11
Purging Audit Files 7-12
ACFS Encryption 7-13
Encrypting ACFS File Systems 7-16
ACFS Replication 7-17
ACFS Replication Requirements 7-19
Managing ACFS Replication 7-21
Using Replication in Conjunction with Cloud FS Security and Encryption 7-24
ACFS Tagging 7-25
Tagging ACFS File Systems 7-26
Generic API for Cloud FS Tagging 7-28
Cloud FS Plug-in Infrastructure 7-29
Using the Cloud FS Plug-in 7-30
High Availability NFS 7-31
Configuring High Availability NFS 7-32
Miscellaneous Cloud FS Enhancements 7-33
Quiz 7-34
Summary 7-36
Practice 7 Overview: Oracle CloudFS Advanced Topics 7-37

# 1

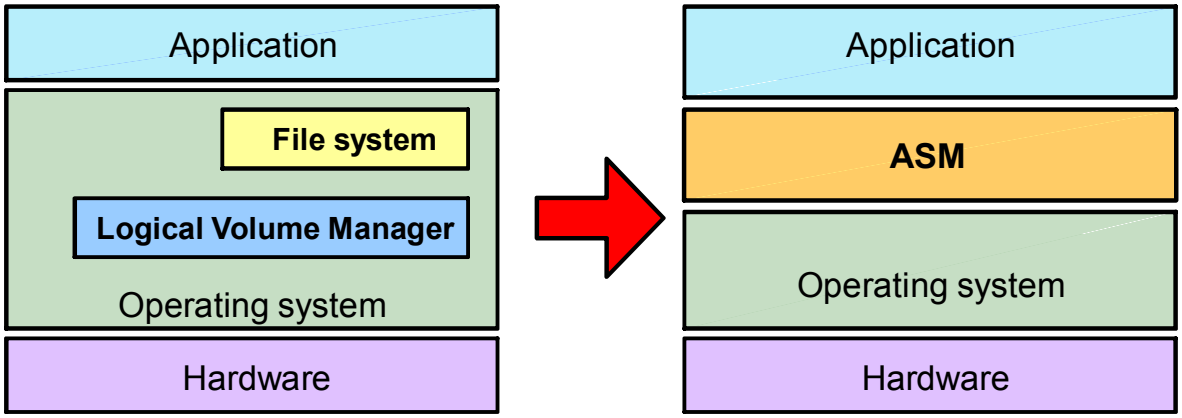
# ASM Overview

# Objectives

After completing this lesson, you should be able to:

- Describe the Automatic Storage Management (ASM) architecture
- Describe the 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..

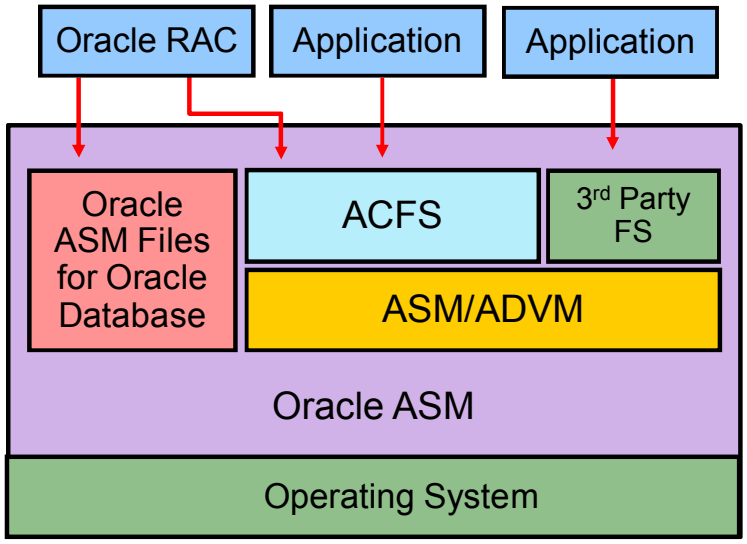
Combining volume management functions with a file system allows a level of integration and efficiency that would not otherwise be possible. For example, ASM is able to avoid the overhead associated with a conventional file system and achieve native raw disk performance for Oracle data files and other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

# ASM CloudFS and ACFS

- ASM manages Oracle database files.
- ACFS manages other files.
- Spreads data across disks to balance load
- Provides integrated mirroring across disks






Oracle Cloud File System (Oracle CloudFS) is designed to help organizations deploy their applications, databases, and storage in private clouds. It delivers a cloud infrastructure that provides network access, rapid elasticity, and provisioning for pooled storage resources that are the key requirements for cloud computing. Customers can use Oracle CloudFS to manage and store all database file types, including general purpose files. Oracle CloudFS includes Oracle Automatic Storage Management Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM).

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. 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 above. 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 Oracle Clusterware technology, participating directly in Clusterware cluster membership state transitions and in Oracle Clusterware resource-based high availability (HA) management. In addition, Oracle installation, configuration, verification, and management tools have been updated to support Oracle ACFS.

# Oracle Flex ASM

- Oracle Flex ASM enables an Oracle ASM instance to run on a separate physical server from the database servers.
- Larger clusters of ASM instances can support more clients while reducing the ASM footprint for the overall system.
- With Flex ASM, you can consolidate all the storage requirements into a single set of disk groups.
  - These disk groups are mounted by and managed by a small set of Oracle ASM instances running in a single cluster.
- ASM clients can be configured with direct access to storage or the I/Os can be sent through a pool of I/O servers.

ORACLE



Oracle Flex ASM enables an Oracle ASM instance to run on a separate physical server from the database servers. With this deployment, larger clusters of Oracle ASM instances can support more database clients while reducing the Oracle ASM footprint for the overall system.

With Oracle Flex ASM, you can consolidate all the storage requirements into a single set of disk groups. All these disk groups are mounted by and managed by a small set of Oracle ASM instances running in a single cluster. You can specify the number of Oracle ASM instances with a cardinality setting. The default is three instances.

When using Oracle Flex ASM, you can configure Oracle ASM clients with direct access to storage or the I/Os can be sent through a pool of I/O servers.

A cluster is a set of nodes that provide group membership services. Each cluster has a name that is globally unique. Every cluster has one or more Hub nodes. The Hub nodes have access to Oracle ASM disks. Every cluster has at least one private network and one public network. If the cluster is going to use Oracle ASM for storage, it has at least one Oracle ASM network. A single network can be used as both a private and an Oracle ASM network. For security reasons, an Oracle ASM network should never be public. There can be only one Oracle Flex ASM configuration running within a cluster.

# ASM Features and Benefits

- Stripes files rather than logical volumes
- Provides redundancy on a file basis
- Enables online disk reconfiguration and dynamic rebalancing
- Reduces the time significantly 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 the Grid Infrastructure

ORACLE



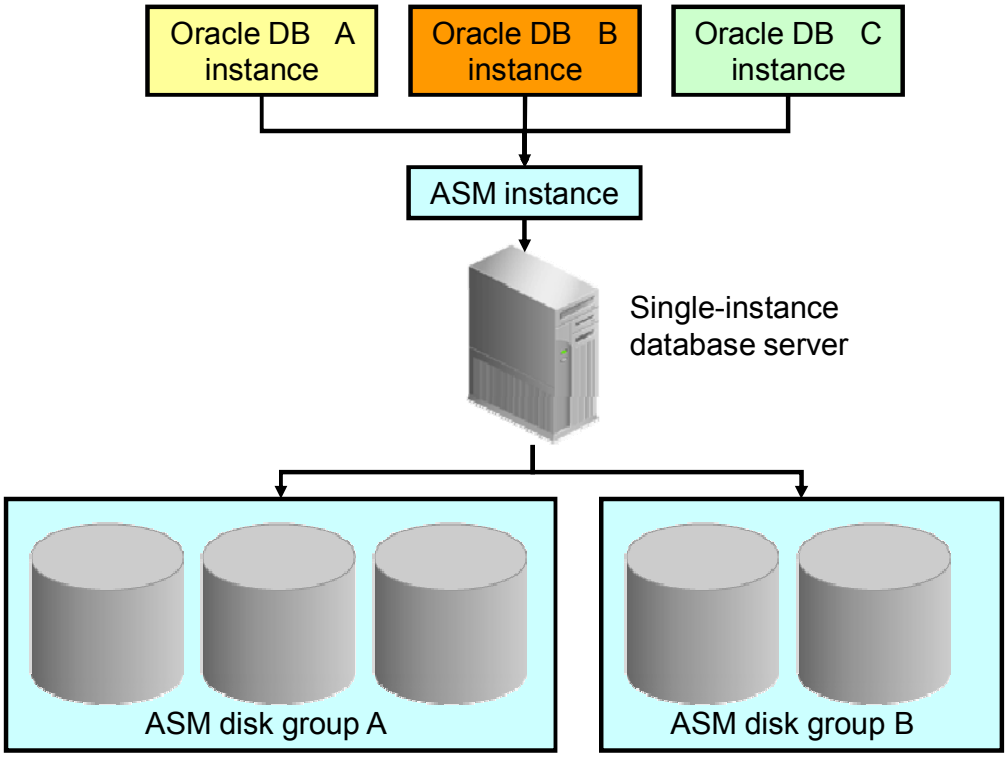
ASM provides striping and mirroring without the need to purchase a third-party Logical Volume Manager. ASM divides a file into pieces and spreads them evenly across all the disks. ASM uses an index technique to track the placement of each piece. Traditional striping techniques use mathematical functions 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 the fast resynchronization of disks that may suffer a transient failure.

ASM supports all Oracle database file types. ASM 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.

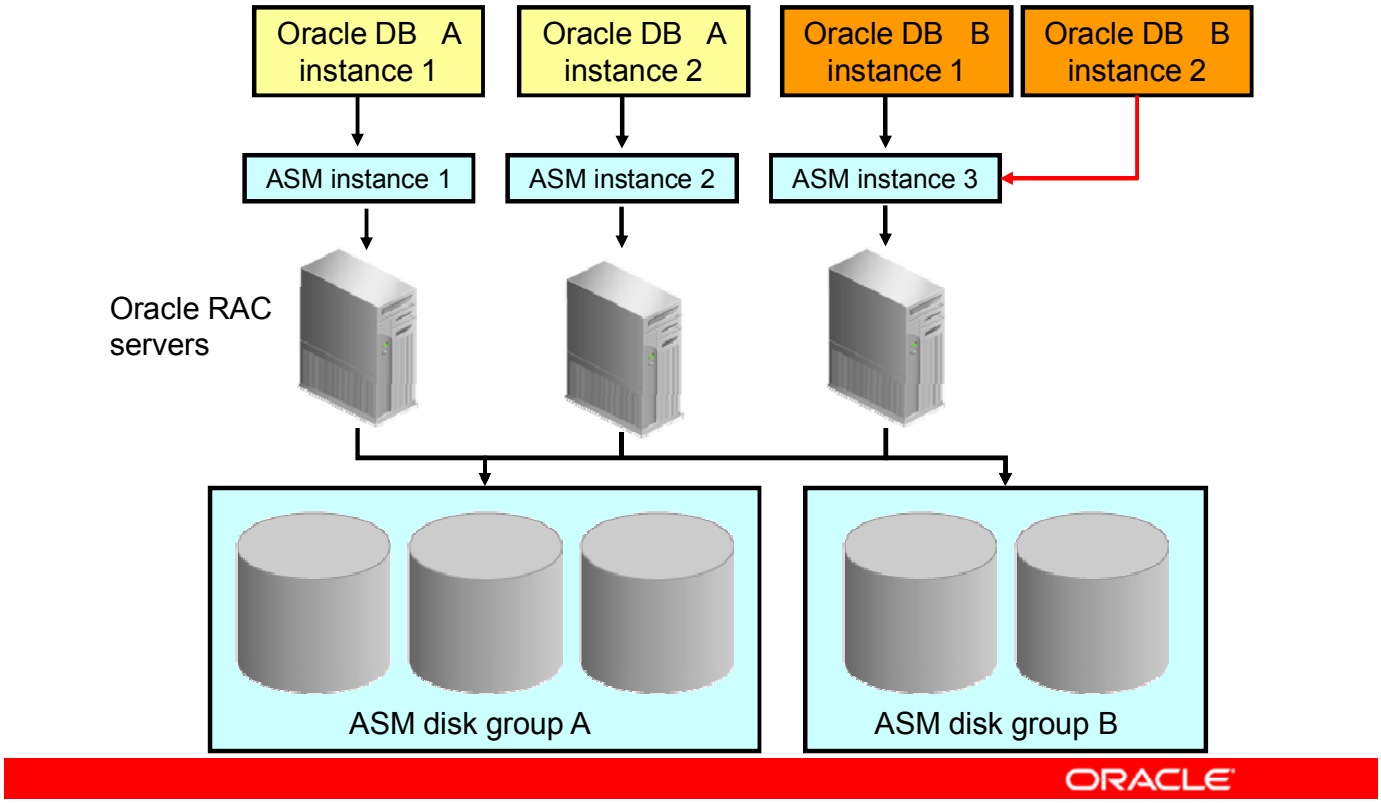
# ASM Instance Designs: Nonclustered ASM and Oracle Databases






This 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.

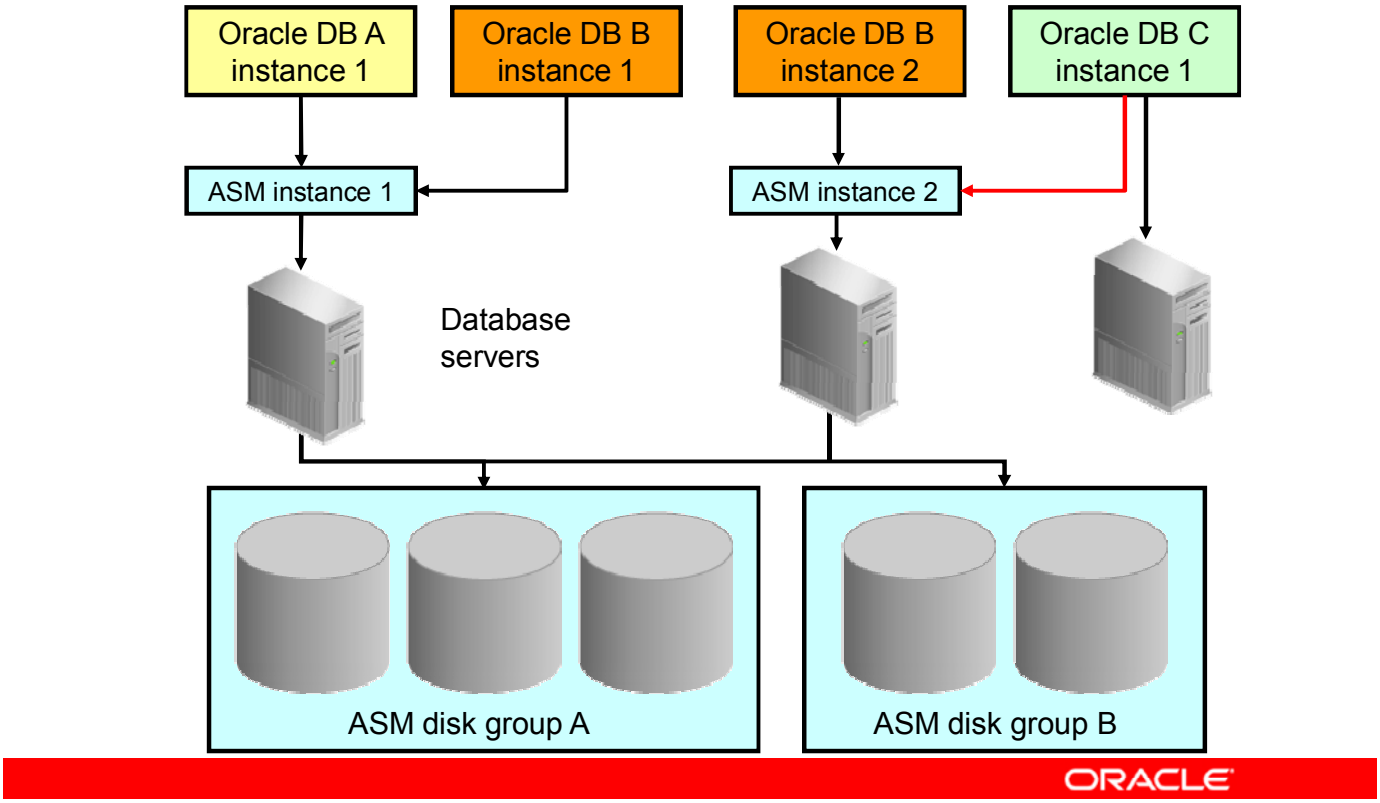
# ASM Instance Designs: Clustered ASM for Clustered Databases






This 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, then it is not necessary for each node hosting a database instance to also host an ASM instance. In the example above, two RAC databases are shown, with Oracle RAC database “A” having instances on the first two nodes, and then Oracle RAC database “B” having 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 shown. In this design, ASM provides storage consolidation, and RAC possibly provides database consolidation.

# ASM Instance Designs: Clustered ASM for Mixed Databases






This 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 above, Oracle Database C, instance 1 is a client of ASM instance 2, incorporating Flex ASM, however the model is still relevant for clusters based on standard ASM.

# ASM Components: Software

For ASM installation of software:

- The directories are located by the operating system environment variables.
  - `ORACLE_BASE` is the top-level directory for a particular software owner.
  - `ORACLE_HOME` is used to identify the top-level directory of the Grid Infrastructure software.
- Use a common `ORACLE_BASE` for all Oracle products owned by the same user.
- Use an isolated `ORACLE_HOME` location from other Oracle products even if they are the same version.
- Do not place Grid `ORACLE_HOME` below `ORACLE_BASE`.



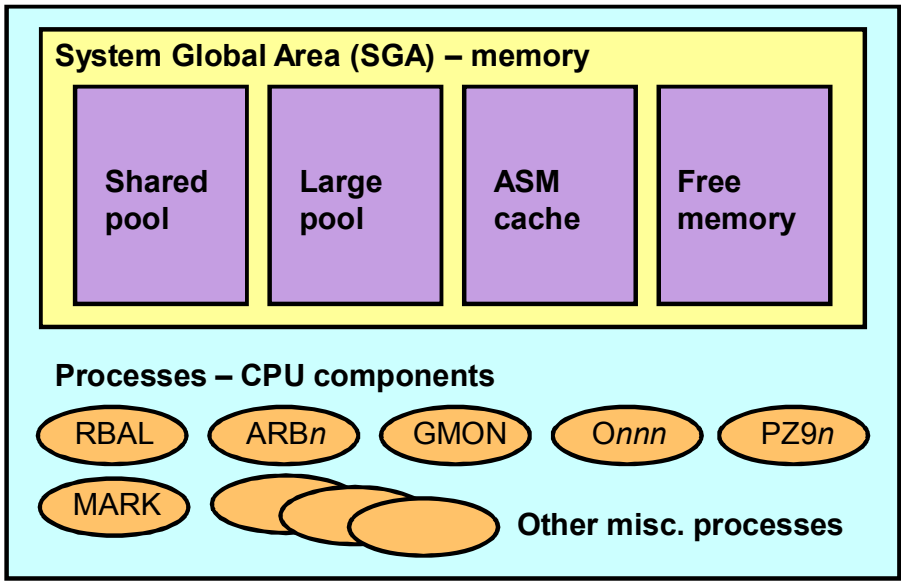
You are required to create two directories for the installation of the software binaries: the Oracle base and the Oracle home directories. The Oracle base directory is a top-level directory for Oracle software installations per software owner. Optimal Flexible Architecture (OFA) guidelines recommend that you use a path similar to `/u01/app/<software owner>`, where the software owner is `oracle` if you use a single owner installation for both Database and Grid Infrastructure, and the owner is `grid` if you use separate owners for Database and Grid Infrastructure software. Create the operating system environment variable `$ORACLE_BASE` to locate the Oracle base directory. If you have set the environment variable `$ORACLE_BASE` for the `grid` user, then the Oracle Universal Installer creates the Oracle Inventory directory in the `$ORACLE_BASE/../oraInventory` path. The Oracle Inventory directory (`oraInventory`) stores an inventory of all software installed on the system.

The Oracle home directory is the directory where you choose to install the software for a particular Oracle product. You must install different Oracle products, or different releases of the same Oracle product, in separate Oracle home directories. OFA guidelines recommend that you use a path similar to `/u01/app/12.1.0/grid`.

Create the operating system environment variable `$ORACLE_HOME` to locate the Oracle home directory for the Grid Infrastructure software.

# 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 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 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 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 (***Jnnn***) to run the jobs)
- ***Jnnn*:** 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, the termination of this process brings down the ASM instance.
- **VBG*n*:** 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). The 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 Components: ASM Instance Primary Processes

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

| Process | Description |
|---|---|
| **RBAL** | Opens all device files as part of discovery and coordinates the rebalance activity |
| **ARB*n*** | One or more slave processes that do the rebalance activity |
| **GMON** | Responsible for managing the disk-level activities such as drop or offline and advancing the ASM disk group compatibility |
| **MARK** | Marks ASM allocation units as stale following a missed write to an offline disk |
| **O*nnn*** | One or more ASM slave processes forming a pool of connections to the ASM instance for exchanging messages |
| **PZ9*n*** | One or more parallel slave processes used in fetching data on clustered ASM installation from GV$ views |

ORACLE



The ASM instance uses dedicated background processes for much of its functionality. The RBAL process coordinates the rebalance activity for disk groups in an Automatic Storage Management instance. It performs a global open on Automatic Storage Management disks. The ARB*n* processes perform the actual rebalance data extent movements in an Automatic Storage Management instance. There can be many of these at a time, called ARB0, ARB1, and so on. The GMON process maintains disk membership in 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 will appear the moment the instance is started, and will disappear after that. They form a pool of connections to the ASM instance for exchanging messages and only appear when needed. The PZ9*n* processes represent one or more parallel slave processes that are used in fetching data when ASM is running in a clustered configuration on more than one machine concurrently.

# ASM Components: Node Listener

The node listener is a process that helps establish network connections from ASM clients to the ASM instance.

- Runs by default from the Grid `$ORACLE_HOME/bin` directory
- Listens on port 1521 by default
- Is the same as a database instance listener
- Is capable of listening for all database instances on the same machine in addition to the ASM instance
- Can run concurrently with separate database listeners or be replaced by a separate database listener
- Is named `tnslsnr` on the Linux platform



The node listener is a process that helps establish network connections from ASM clients to the ASM instance. The listener process is installed with the ASM software installation. It runs by default from the Grid `$ORACLE_HOME/bin` directory on port 1521. The node listener is the same as a database listener or Oracle Net Listener. Because ASM is usually installed before installing the database, the node listener is configured and started automatically to provide a gateway to the ASM instance for remote clients. This listener is capable of listening for all instances on that machine, including database instances. After the ASM installation, a database installation is performed into a different `ORACLE_HOME` location. Each `ORACLE_HOME` location can provide additional listener processes and a set of configuration files. The node listener and separate database listeners can run concurrently on different port numbers or the node listener can be replaced by a separate database listener. There are two configuration files for the listener process. They are as follows:

`$ORACLE_HOME/network/admin/listener.ora`

`$ORACLE_HOME/network/admin/sqlnet.ora (Optional)`

The listener process is named `tnslsnr` on the Linux platform. It can be managed with the `srvctl` utility. The syntax to start and stop the listener is as follows:

```
srvctl start listener -n nodename
srvctl stop listener -n nodename
```

# ASM Components: Configuration Files

The ASM installation of software uses several configuration files to define the environment.

- ASM instance configuration files include:
  - The server parameter file (`SPFILE`), which initializes the ASM instance and defines startup parameters
  - `orapw+ASM`, which is the binary password file used for remote authentication to the ASM instance
- Node listener configuration files include:
  - `listener.ora`, a text file that defines the node listener
  - `sqlnet.ora`, an optional text file that provides additional listener options
- Other miscellaneous text configuration files include:
  - `/etc/oratab`, which lists all the instances on the host machine
  - `/etc/oraInst.loc`, which defines the Oracle inventory directory

ORACLE



Most of the metadata that relates to ASM is stored within the ASM disk group and in the memory-based tables of the ASM instance. A few other files are used for configuration. The ASM instance uses two configuration files. The server parameter file that defines initialization parameters at startup is a binary file created at installation by default in a specified ASM disk group at `+<disk_group>/<cluster>/ASMPARAMETERFILE/register.253.`*nnnnnn*. The administrator may create a text initialization parameter file by default named `init+ASM`*n*`.ora`. The text and binary versions are mutually exclusive for startup.

The other file that is used to configure the instance is the binary password file named `orapw+ASM`. This file is used for local and remote authentication from ASM clients. The password file can be found in the `$ORACLE_HOME/dbs` directory.

The node listener also uses two configuration files. The main configuration file for the listener is the text file `listener.ora`. This defines the protocol along with protocol-specific information for listening connection endpoints. The other file is the optional text file `sqlnet.ora`. It is used to define the computer's domain and default resolution method. Both these configuration files can be found in the `$ORACLE_HOME/network/admin` directory.

There are a few other miscellaneous text configuration files. The `/etc/oratab` text file lists all the instances on the host machine and the `/etc/oraInst.loc` text file defines the location of the central Oracle inventory.

# ASM Components: Group Services

Group services provided by Oracle Clusterware allow for cooperating applications to communicate in a peer environment.

- Group services for the Oracle environment:
  - Provide information necessary to establish connections
  - Provide assistance in doing lock recovery
  - Guarantee ASM disk group number uniqueness
  - Monitor node membership, evictions, and cluster locks
- Oracle Clusterware is responsible for:
  - Starting and stopping ASM instances
  - Mounting and dismounting disk groups
  - Mounting and dismounting ACFS volumes
  - Starting and stopping dependent database instances

ORACLE



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

- Enable the database instance to locate the ASM instance along with credentials to the ASM instance for establishing an Oracle Call Interface (OCI) connection
- Provide assistance in doing 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 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 node membership and heartbeat monitoring, basic cluster locking, and node evictions.

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

# 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

ORACLE



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 the 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 belonging 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 Disk Group: 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.
- The 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 the failure group name 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 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 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 added to a disk group 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 | OCR and Voting files |
| 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 database name 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 the 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.
- The data extent size is:
  - 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 use 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 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. It will increase by a factor of 16 after 40,000 extents, where it will remain 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. The coarse-grained stripes are always equal to the effective AU size. The 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. The 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: 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 will allocate the 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<br>Three-way<br>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), ASM 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. The 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 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, this worst failure refers to a permanent failure where the disks must be dropped, not the case where the disks go offline and then 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 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, then the value would be the size of the largest capacity disk.
- High redundancy disk group with more than three failure groups

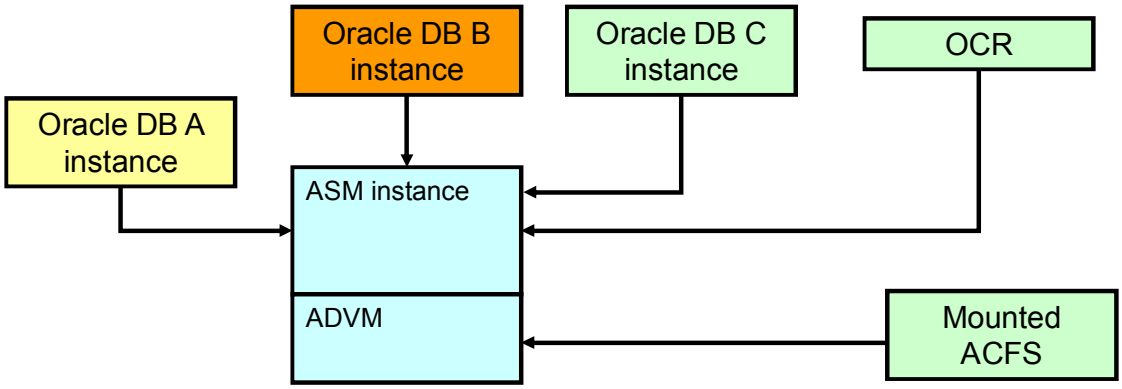
The value is the total raw space for all of the disks in the two largest failure groups.

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 equal to twice their size, if 4 GB of actual usable file space remains, then `USABLE_FILE_MB` equals roughly 2 GB. You can then add a file that is up to 2 GB.

# ASM Components: ASM Clients

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

- Connected ASM clients can be viewed:
  - 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 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 being used.

An OCR and voting file stored in ASM is 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 will be referred to as ASM utilities in this course.

# ASM Components: 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`)



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 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.

# 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<br>Create and delete files.<br>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.

ORACLE



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.

# ASM OS Groups with Role Separation

To separate the duties of ASM administrators and DBAs, there are six OS groups:

| Group | For | Example OS Group | Privilege |
|---|---|---|---|
| OSASM | ASM | asmadmin | SYSASM |
| OSDBA | ASM | asmdba | SYSDBA |
| OSOPER | ASM | asmoper | SYSOPER |
| oraInventory group | Both | oinstall | |
| OSDBA | DB | dba | SYSDBA |
| OSOPER | DB | oper | SYSOPER |



The OSASM group is used exclusively for ASM. Members of this group can connect as `SYSASM` using OS authentication and have full access to ASM. Initially, only the OS user that installs the Grid Infrastructure software is a member of this group. However, other users can be added. This group also owns the ASM disks. The Oracle binaries must `setgid` to this group, so that Oracle instances and tools can have access to ASM disks without giving users direct access to these devices. For this example, the name of this group is `asmadmin`.

In this example, the OSDBA group for ASM is called `asmdba`. Members of this group can connect to the ASM instance as `SYSDBA`. Every database instance owner that will use ASM for storage must be a member of this group. The OSDBA group for ASM could be a different group than the OSDBA group for a database.

In this example, the OSOPER group for ASM is called `asmoper`. Members of this OS group have the `SYSOPER` privilege, which limits the set of allowable SQL commands to the minimum required for the basic operation of an already configured ASM instance. The OSOPER group for a database would be a different group.

# Authentication for Accessing ASM Instances

There are three modes of connecting to ASM instances:

- Local connection using operating system authentication

```
$ sqlplus / AS SYSASM
```

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

- Local connection using password file authentication

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

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

- Remote connection using Oracle Net Services and password authentication

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

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



There are three modes of connecting to ASM instances:

- **Local connection 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 using:

  ```
  CONNECT / AS SYSASM
  ```

  **Note:** A local connection using `AS SYSDBA` always uses OS authentication even when using the username/password syntax.
- **Local connection using password file authentication:** The following example shows a local connection using password file authentication:

  ```
  CONNECT sys/<sys_password> AS SYSASM
  ```
- **Remote connection by way of Oracle Net Services using password authentication:** Password-based authentication is also supported remotely using Oracle Net Services. The following example shows a remote connection by way of Oracle Net Services 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 which 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 to 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 to 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 using:
  - SQL*Plus GRANT command
  - 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 the Oracle Universal Installer. You can create a password file for ASM manually 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
```

# Using a Single OS Group

| Role/Software | Software Owner | 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 in this case are the primary group for the `oracle` user `oinstall` and the secondary group for the `oracle` user `dba`.

The disk devices will be 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

| Role/Software | Software Owner | Groups/Privileges/OS Group |
|---|---|---|
| Oracle ASM administrator<br>Oracle Grid Infrastructure home | grid | asmadmin (OSASM)/SYSASM<br>asmdba (OSDBA for ASM)/SYSDBA<br>asmoper (OSOPER for ASM)/SYSOPER<br>dba1, dba2, ... (OSDBA for the databases when in an Oracle Restart configuration) |
| Database administrator 1<br>Database home 1 | oracle1 | asmdba (OSDBA for ASM)/SYSDBA<br>oper1 (OSOPER for database 1)/SYSOPER<br>dba1 (OSDBA for database 1)/SYSDBA |
| Database administrator 2<br>Database home 2 | oracle2 | asmdba (OSDBA for ASM)/SYSDBA<br>oper2 (OSOPER for database 2)/SYSOPER<br>dba2 (OSDBA for database 2)/SYSDBA |
| Operating system disk device owner | grid | asmadmin (OSASM) |

ORACLE



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 and different groups. The database administrators of database1 are different from the administrators of database2.

# ASM Scalability

ASM has the following storage limits if the `COMPATIBLE.ASM` disk group attribute is set to 12.1 or greater:

- 511 disk groups in a storage system
- 10,000 ASM disks in a storage system
- For each ASM disk:
  - 4 PB maximum storage for each ASM disk with an AU of 1 MB
  - 8 PB maximum storage for each ASM disk with an AU of 2 MB
  - 16 PB maximum storage for each ASM disk with an AU of 4 MB
  - 32 PB maximum storage for each ASM disk with an AU of 8 MB
- 320 exabyte maximum storage for each storage system
- 1 million files for each disk group
- ASM file size limits (DB limit is 128 TB):
  - External redundancy maximum file size is 140 PB.
  - Normal redundancy maximum file size is 23 PB.
  - High redundancy maximum file size is 15 PB.

ORACLE



ASM imposes the following limits:

- 511 disk groups in a storage system
- 10,000 ASM disks in a storage system
- For each ASM disk:
  - 4 petabyte maximum storage for each ASM disk with an AU of 1 MB
  - 8 petabyte maximum storage for each ASM disk with an AU of 2 MB
  - 16 petabyte maximum storage for each ASM disk with an AU of 4 MB
  - 32 petabyte maximum storage for each ASM disk with an AU of 8 MB
- 320 exabyte maximum storage for each storage system
- 1 million files for each disk group

The Oracle database supports data file sizes up to 128 TB. ASM supports file sizes greater than 128 TB in any redundancy mode. This provides near-unlimited capacity for future growth. The ASM file size limits when `COMPATIBLE.RDBMS` >= 11.1 as follows:

- External redundancy: 140 petabytes
- Normal redundancy: 23 petabytes
- High redundancy: 15 petabytes

**Note:** In non-Exadata environments, ASM disks larger than 2 terabytes require `COMPATIBLE.ASM` set to 12.1 and only12.1 database instances are supported.

Some standard metric prefixes include the following:

- 1 gigabyte = 1,000,000,000 bytes or $10^9$ bytes using a decimal prefix
- 1 terabyte = 1,000,000,000,000 bytes or $10^{12}$ bytes using a decimal prefix
- 1 petabyte = 1,000,000,000,000,000 bytes or $10^{15}$ bytes using a decimal prefix
- 1 exabyte = 1,000,000,000,000,000,000 bytes or $10^{18}$ bytes using a decimal prefix

# 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 Automatic Storage Management (ASM) architecture
- Describe the components of ASM

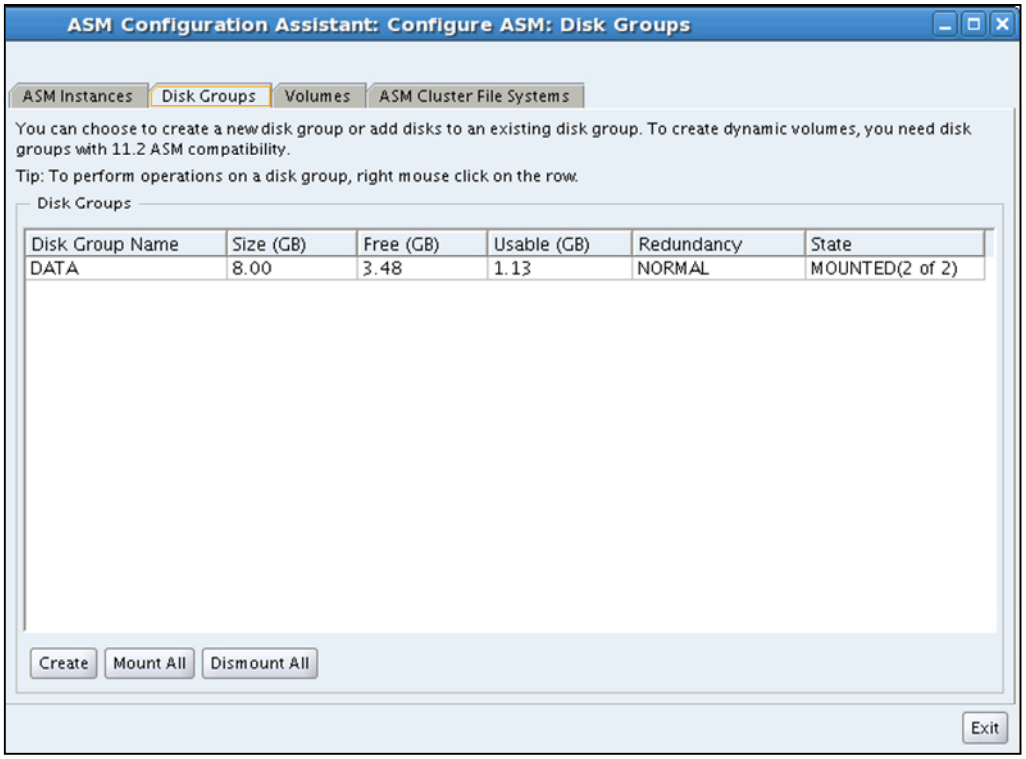
# 2 Administering ASM Instances

# 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

# Managing ASM with ASMCA






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 assuming that you are connected as the `grid` software owner on the first node of your cluster:

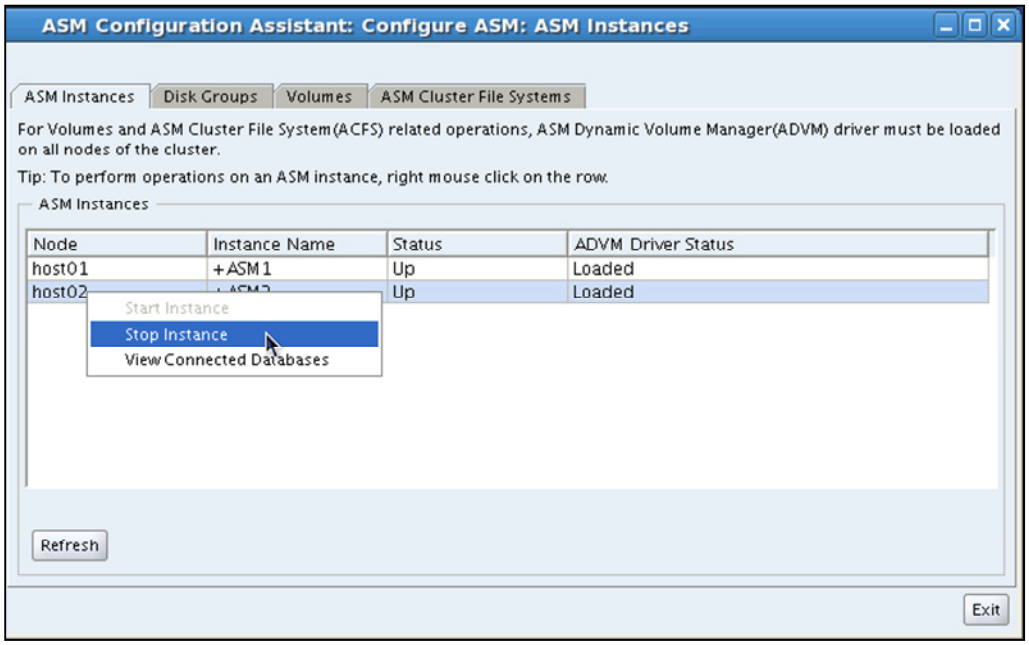
```
$ . oraenv
ORACLE_SID = [+ASM1] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/12.1.0/grid is
/u01/app/grid
```

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 Using ASMCA






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

# Starting and Stopping ASM Instances Using ASMCMD

- Use ASMCMD to shut down an Oracle ASM instance.
  - Shutdown normally:

```
ASMCMD> shutdown [--normal]
```

  - Shutdown, aborting current operations:

```
ASMCMD> shutdown [--abort]
```

  - Shutdown immediately:

```
ASMCMD> shutdown [--immediate]
```

- Use ASMCMD to start an Oracle ASM instance.
  - Start instance normally:

```
ASMCMD> startup
```

  - Start using the `asm_init.ora` parameter file:

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

  - Start in restricted mode:

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

ORACLE



The ASMCMD utility includes the ability to start and stop ASM instances. The following are examples of the shutdown command. The first example performs a shut down of the Oracle ASM instance with normal action. The second example performs a shut down with immediate action. The third example performs a shut down 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 no mount 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 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.
$
```

ORACLE



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 *`Grid_home`*`/bin` location established during the 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 hosting the ASM instance (optional); if omitted, the ASM instance will be started on all nodes.
  - *`<asm_inst>`* is the name of the specific ASM instance 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 using SQL*Plus is similar to the way in which you start and stop database instances.

```
$ export ORACLE_SID=+ASM1
$ export ORACLE_HOME=/u01/app/12.1.0/grid
$ $ORACLE_HOME/bin/sqlplus / AS SYSASM

SQL*Plus: Release 12.1.0.1.0 Production on Tue Jul 30 19:56:51 2013

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 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, similar to the way in which 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 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 files that are in that disk group. Also, if a disk group is mounted by an instance in restricted mode, that 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. Rebalance operations that occur while a disk group is in restricted mode eliminate the lock and unlock extent map messaging that occurs between 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 database instances that use the ASM instance before attempting to shut down the ASM instance.

The following list describes the SHUTDOWN command parameters 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 Oracle ASM Dynamic Volume Manager (ADVM) volumes, those file systems should first be dismounted.

# Starting and Stopping ASM Instances Containing Cluster Files

ASM instances containing the OCR and voting disks:

- Will be automatically restarted by the high availability services daemon
- Use `crsctl stop crs`



ASM instances that contain the OCR files or voting disk files will be automatically restarted by the high availability services daemon (OHASD). ASM instances with connected clients cannot be shut down except with the `abort` option.

Oracle Clusterware is a client of ASM when the OCR files and voting files are in ASM disk groups. Stopping the Oracle Clusterware services includes stopping ASM.

# ASM Initialization Parameters

- ASM initialization parameters can be set:
  - With Oracle ASM Configuration Assistant (ASMCA).
  - After installation, by using `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 SQL `ALTER SYSTEM` or `ALTER SESSION` 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 the 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. The 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: the location set in the Grid Plug and Play profile, then for an SPFILE in the Oracle home, 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 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 SQLPLUS 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 groups 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
  - Meaning ASM searches a default path looking for disks that it has read-and-write access to
    - Default search path is platform specific.
    - Default search path includes Oracle ASMLib disks.
- Can use * and ? as wildcards and regular expression syntax.
- Can be set dynamically using ALTER SYSTEM
  - A change will be rejected if the proposed value cannot be used to discover *all* the disks belonging to the currently mounted disk groups.

ORACLE



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 discovery string format 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 the default pattern matching.

For example, on a Linux server that does not use ASMLib, to limit the discovery process to only include 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 only include 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 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 of 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 disks must be discoverable by all of the nodes in the cluster. This may require dynamically changing the initialization parameter to enable adding new storage.

**Note:** ASM cannot use a disk unless all the ASM instances in the cluster can discover the disk through one of their own discovery strings.

# ASM_POWER_LIMIT

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

- Default is 1.
  - Meaning ASM conducts rebalancing operations using minimal system resources
- 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 using ALTER SYSTEM or ALTER SESSION.



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 for the ASM_POWER_LIMIT initialization parameter. For Oracle Database 11*g Release 2 (11.2.0.1)*, or older, the range of values for ASM_POWER_LIMIT is 0 to 11. If the value of the POWER clause is specified larger than 11 for a disk group with ASM compatibility set to less than 11.2.0.2, then a warning is displayed and a POWER value equal to 11 is used for rebalancing.

The specification of the power of the rebalancing operation in a disk group only affects rebalance operations, not 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 used for MEMORY_TARGET is acceptable for most environments.
- The minimum value for MEMORY_TARGET is 1 GB.
- It can be increased dynamically using ALTER SYSTEM.



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 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 the 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 using a text editor.
- Use the `ALTER SYSTEM` SQL command to adjust 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 using a text editor.

You can use the `ALTER SYSTEM` SQL command to adjust 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.

# 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`



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 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 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 accessible across the cluster reduces maintenance costs and situations where passwords become out of sync.

You can use a password file 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 on 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 on 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 using a password file from a previous release. For example:

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

`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/12.1.0/grid
Password file: +DATA/orapwASM
ASM listener: LISTENER
ASM instance count: 3
Cluster ASM listener: ASMNET1LSNR_ASM
```

# Starting and Stopping the Node Listener

- Using the `lsnrctl` utility:

```
$ lsnrctl start listener
LSNRCTL for Linux: Version 12.1.0.1.0-Production on 31-JUL-2013 16:45:35
Copyright (c) 1991, 2013, Oracle.  All rights reserved.
Starting /u01/app/12.1.0/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 using the `lsnrctl` utility 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 will start a listener with the configuration files in the *Grid_home*. By default, the database installation will use 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 configuration files in the RDBMS home, set `ORACLE_HOME` and `PATH` environment variables to use the RDBMS home and then run the utilities.

# ASM Dynamic Performance Views

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

- Is accessed by ASM utilities to retrieve metadata-only information using the SQL language
- 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.



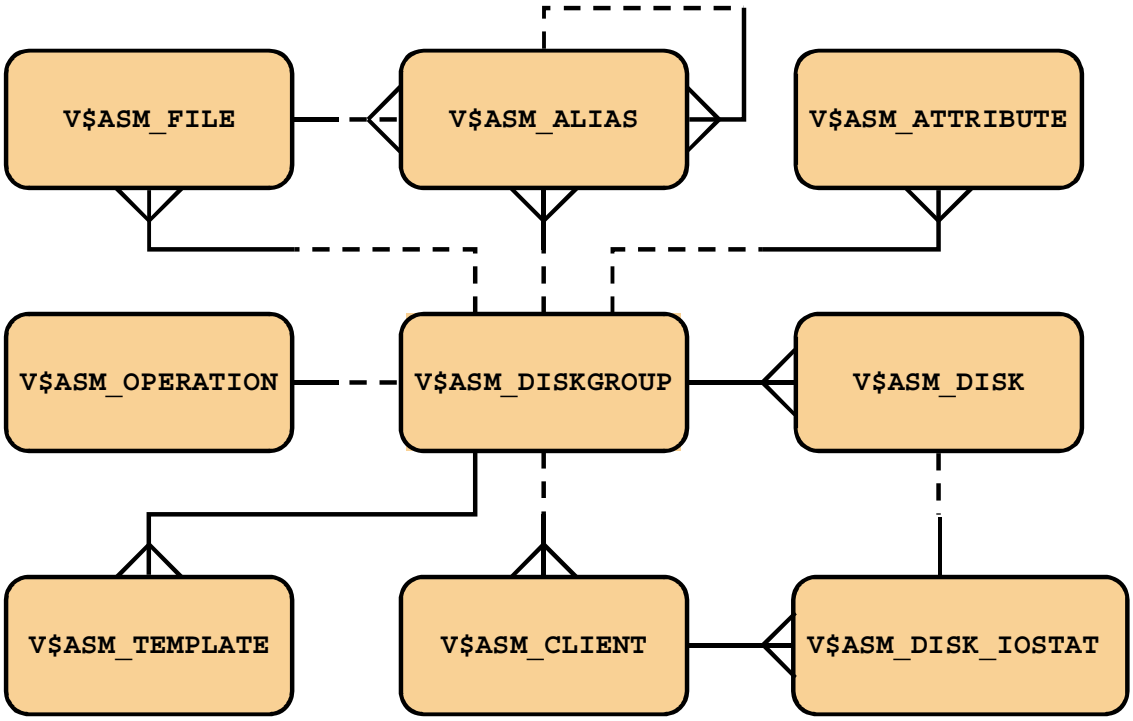
An ASM instance does not have a data dictionary. Instead, information relating to its configuration and operation is available by querying a set of dynamic performance views. The ASM dynamic performance views have names having the prefix V$ASM. They can be queried using SQL in the same manner as any 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 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 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

The recommended configuration for ASM instance initialization is:

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

# 3

# Flex ASM

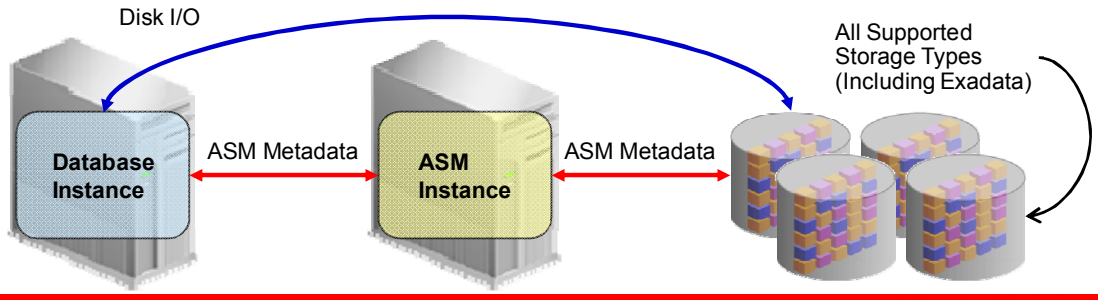
# Objectives

After completing this lesson, you should be able to:

- Describe the architecture and components of Flex ASM
- Install and configure Flex ASM
- Administer Flex ASM

# Flex ASM: Overview

- In previous versions, ASM clients can only access ASM 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.






Prior to Oracle Database 12*c*, an ASM client (database instance or ASM Cluster File System [ACFS]) can only connect to an ASM instance 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.

Note that ASM continues to support the same architecture as previous releases where clients are coupled with ASM instances on the same host. This mode of deployment is called standard ASM.

# Flex ASM and Flex Clusters

- Flex Clusters requires 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



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 of 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.

# 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 number of instances across the cluster.
  - Default is 3.
- All disk groups are mounted by all ASM instances.

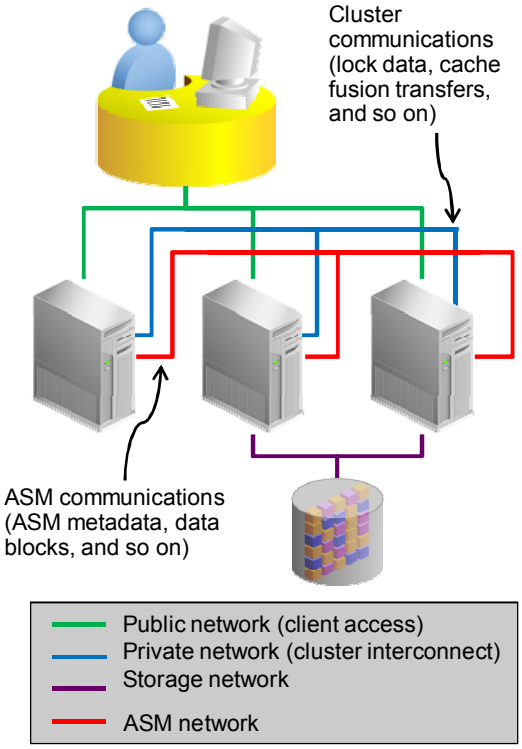


Flex ASM relaxes the requirement for an ASM instance to exist on every node in the cluster. Instead, the administrator specifies the cardinality for ASM. This number specifies the number of ASM instances that should be made available in the ASM cluster. The default cardinality setting for ASM instances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

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

# ASM Network

- In previous 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.





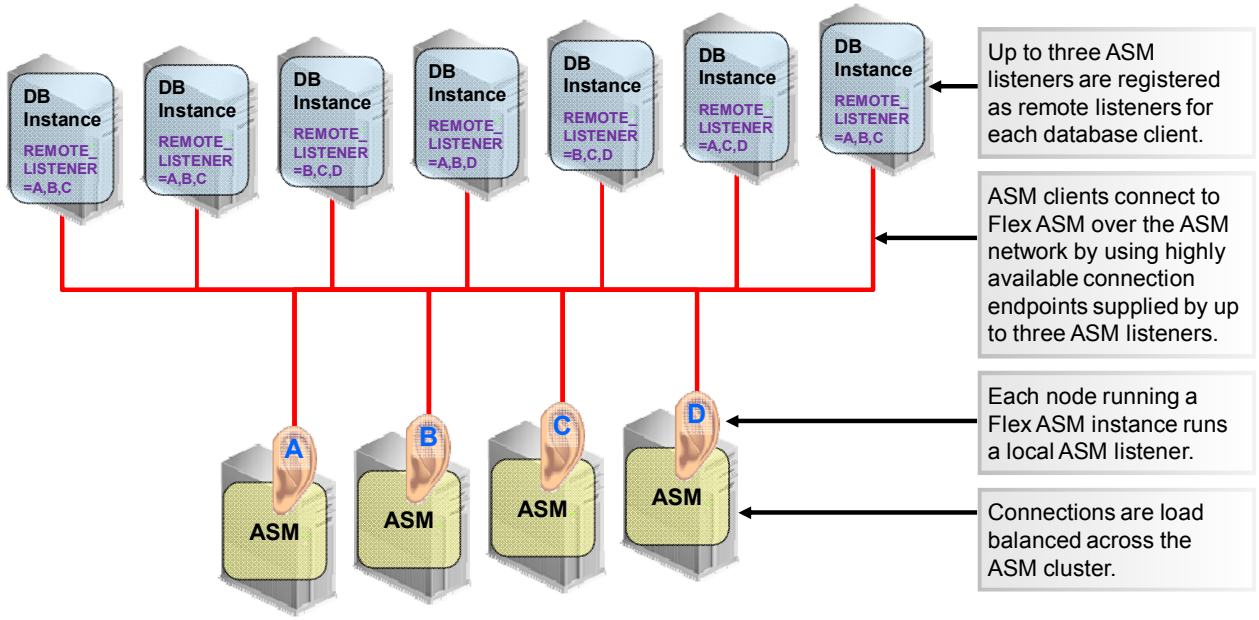

In previous versions, a CSS cluster requires access to a public network and one or more private networks. Clients outside the cluster use the public network to connect to servers in the cluster. The private networks are predominantly used for inter-node communication within the cluster. Sometimes, the 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, and this requires all ASM networks to be accessible on all the nodes hosting 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.

# ASM Listeners






To support Flex ASM connectivity, a set of ASM listeners are 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 ASM listeners.

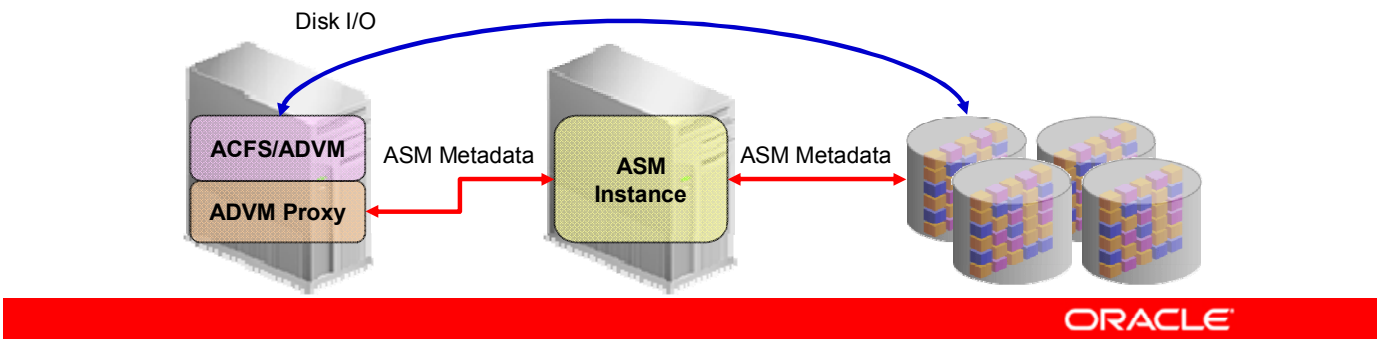
Each node hosting an ASM instance hosts one local ASM listener for each ASM network. Each ASM listener can service client connections over the corresponding ASM network. Up to three ASM listener addresses are registered as remote listeners in each client database instance. Using this arrangement, clients 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.

# 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
- By default, it is configured on every node in a standard cluster or every Hub Node in a Flex Cluster
- It can be shut down when ACFS isn't running



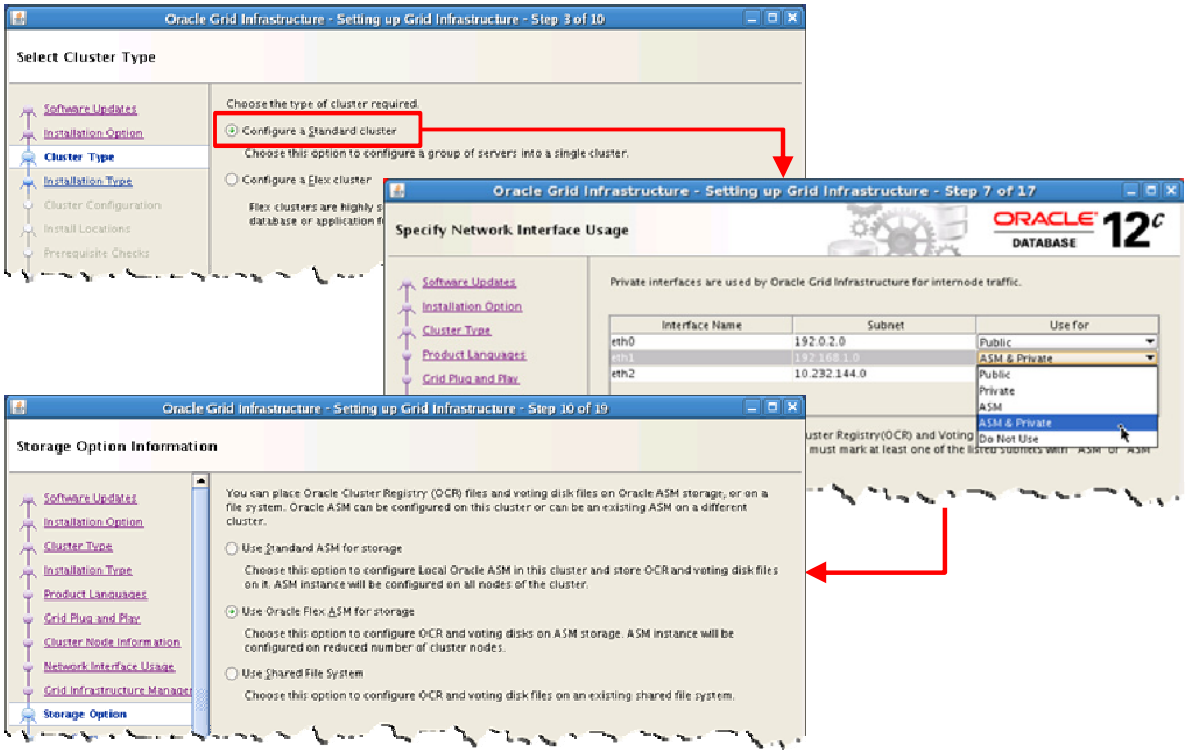



The ADVM Proxy is a special Oracle instance. It's sole purpose is to enable ASM Dynamic Volume Manager (ADVM), and through it ASM Cluster File System (ACFS), to connect to Flex ASM.

In release 12.1, 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. 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>`.

# Configuring Flex ASM on a Standard Cluster



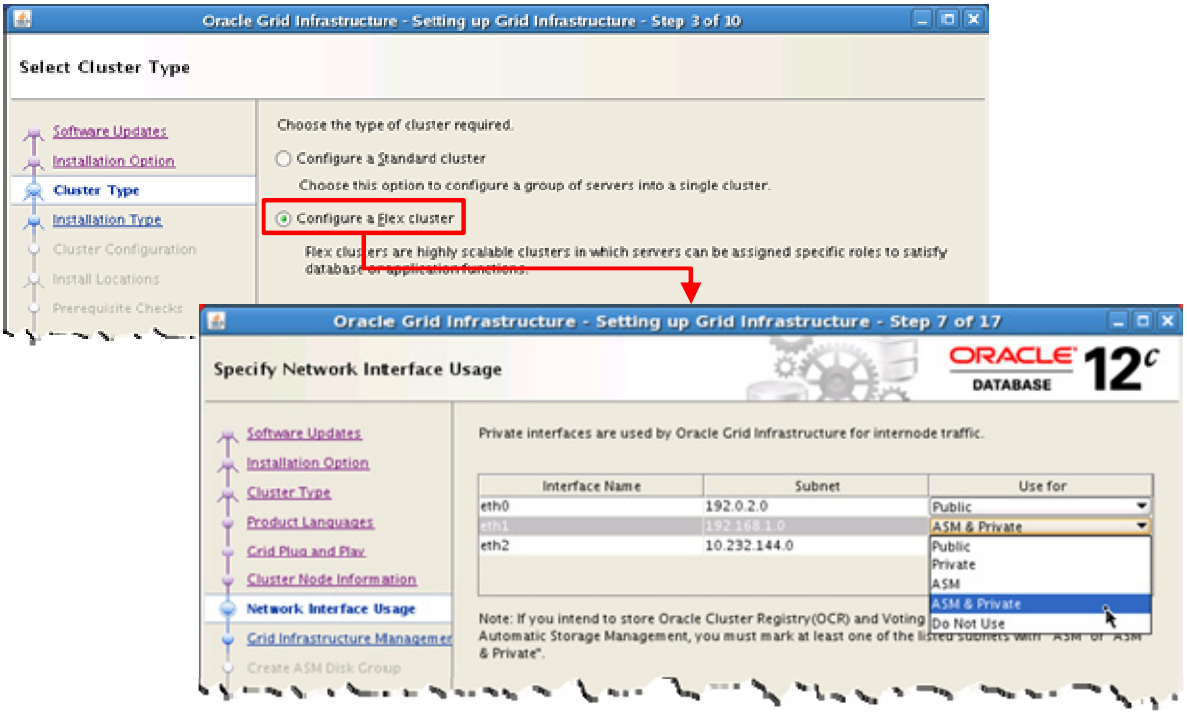



Oracle Universal Installer (OUI) has been updated to facilitate configuration of Flex ASM. The procedure for configuring Flex ASM using OUI differs slightly, depending on whether or not the cluster is also being configured as a Flex Cluster.

To configure Flex ASM on a standard cluster, the following steps are required:

1. Select “Configure a Standard cluster” on the Select Cluster Type screen.
2. Specify an ASM network on the Specify Network Interface Usage screen.
3. Select “Use Oracle Flex ASM for storage” on the Storage Option Information screen.

# Configuring Flex ASM on a Flex Cluster






If the option to configure a Flex Cluster is selected on the Select Cluster Type screen, Flex ASM is implicitly configured and you must specify an ASM network on the Specify Network Interface Usage screen.

# Managing Flex ASM Instances

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

- Primary concern is that instances are up and running.

```
$ srvctl status asm -detail
ASM is running on host03,host02,host01
ASM is enabled.
$ srvctl status asm -proxy -detail
ADVM Proxy is running on host04,host03,host02,host01
ADVM Proxy is enabled.
```

- No Flex ASM-specific instance parameters are required.
- Default settings will effectively support most situations.
- ASM and ADVM Proxy instances use automatic memory management.
  - Minimum default setting: `MEMORY_TARGET=1076M`



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 12.1, 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 12.1, the default setting for `MEMORY_TARGET` is based on various attributes of the node hosting the instance, such as the physical memory size and the number of processor cores.

Note that the minimum default `MEMORY_TARGET` setting (`1076M`) is significantly larger than the default `MEMORY_TARGET` setting used by ASM instances in previous versions.

# Stopping, Starting, and Relocating Flex ASM Instances

- ASM Instances

```
$ srvctl status asm -detail
ASM is running on host03,host02,host01
ASM is enabled.
$ srvctl stop asm -node host03 -f
$ srvctl start asm -node host04
$ srvctl status asm -detail
ASM is running on host04,host02,host01
ASM is enabled.
$ srvctl relocate asm -currentnode host04 -targetnode host03
$ srvctl status asm -detail
ASM is running on host03,host02,host01
ASM is enabled.
```

- ADVM Proxy Instances

```
$ srvctl stop asm -proxy -node host03
$ srvctl start asm -proxy -node host04
```



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
```



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 on 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/12.1.0/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
```



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.
- Clients can be manually relocated 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

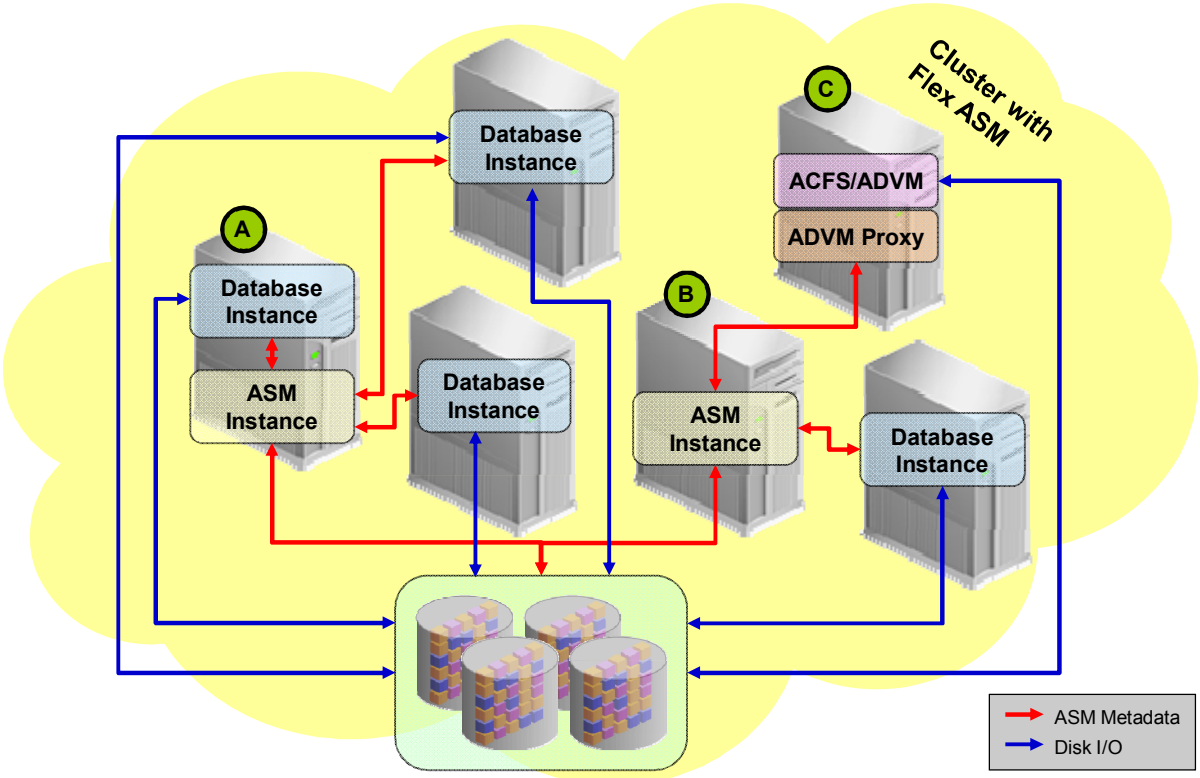


With Flex ASM, if an ASM instance fails, clients are automatically relocated to another instance. When the failure is detected by the client, it reconnects to an available instance. Like any other connection request, reconnection requests are subject to connection load balancing. Relocation due to failure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

In addition to automatic relocation, the `ALTER SYSTEM RELOCATE CLIENT` command can be used on the node 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 and 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 running Flex ASM. The diagram also illustrates the Hub Nodes of a Flex Cluster.
- ASM clients can run on any node in a standard cluster, or any Hub Node in a Flex Cluster. In release 12.1, Flex ASM does not support clients on Leaf Nodes.
- 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 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 separate from the ASM clients (as shown on node B in the diagram).
- By default, the ADVM proxy runs on every node in a standard cluster or every Hub Node in a Flex Cluster. For the sake of simplicity the ADVM proxy is only shown on node C in the diagram, which in this example is the only node running an ASM Cluster File System.

# Quiz

Identify the correct statements regarding server Flex ASM:

a. Flex ASM requires an ASM instance on each cluster node 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.

ORACLE



**Answer: b, c, d**

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

# Quiz

If OUI is used to install and configure a four node standard 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.

ORACLE



**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 3 nodes in a standard cluster, or fewer than 3 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

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.



**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
- Install and configure Flex ASM
- Administer Flex ASM

# Practice 3 Overview: Database Fail Over with Flex ASM

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

# 4

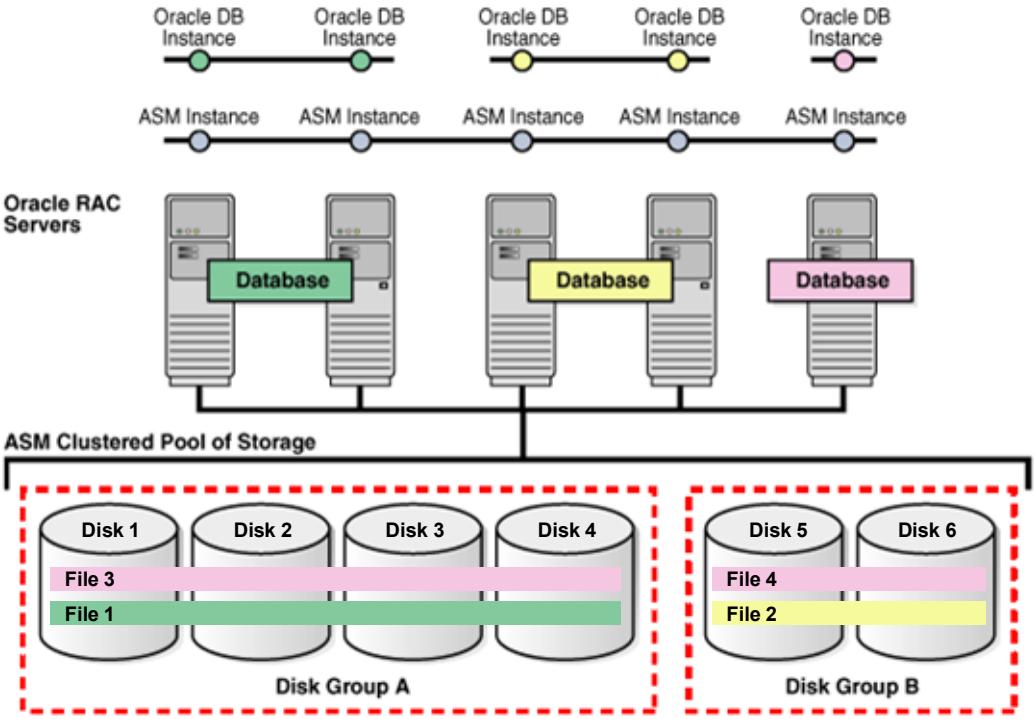
# Administering ASM Diskgroups

# 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
- Explain key performance and scalability considerations for 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 required for the management of space in the disk group. An 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 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 will be 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 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 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="12.1"/>
   <a name="compatible.rdbms" value="12.1"/>
</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.

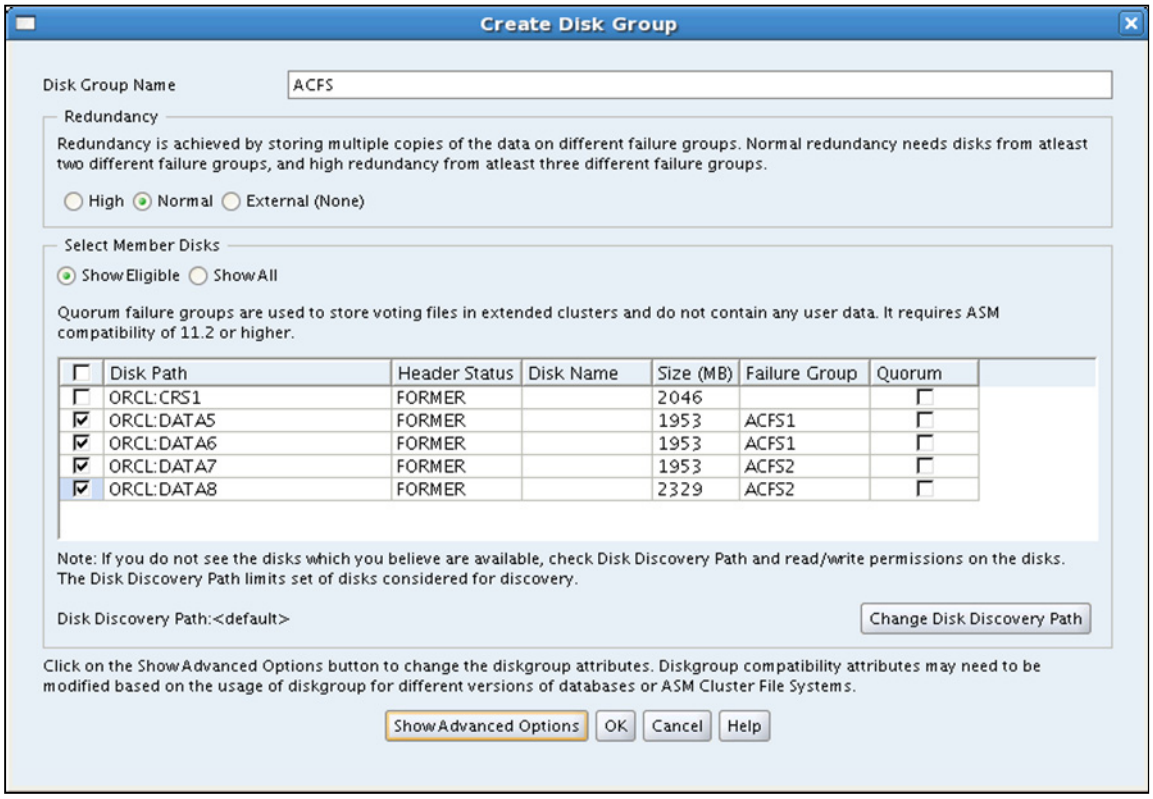
The XML file for the `mkdg` command specifies the name of the disk group, redundancy, attributes, and paths of the disks that form the disk group. Redundancy is an optional parameter; the default is normal redundancy. For some types of redundancy, disks are required to be gathered into failure groups. In the case that 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

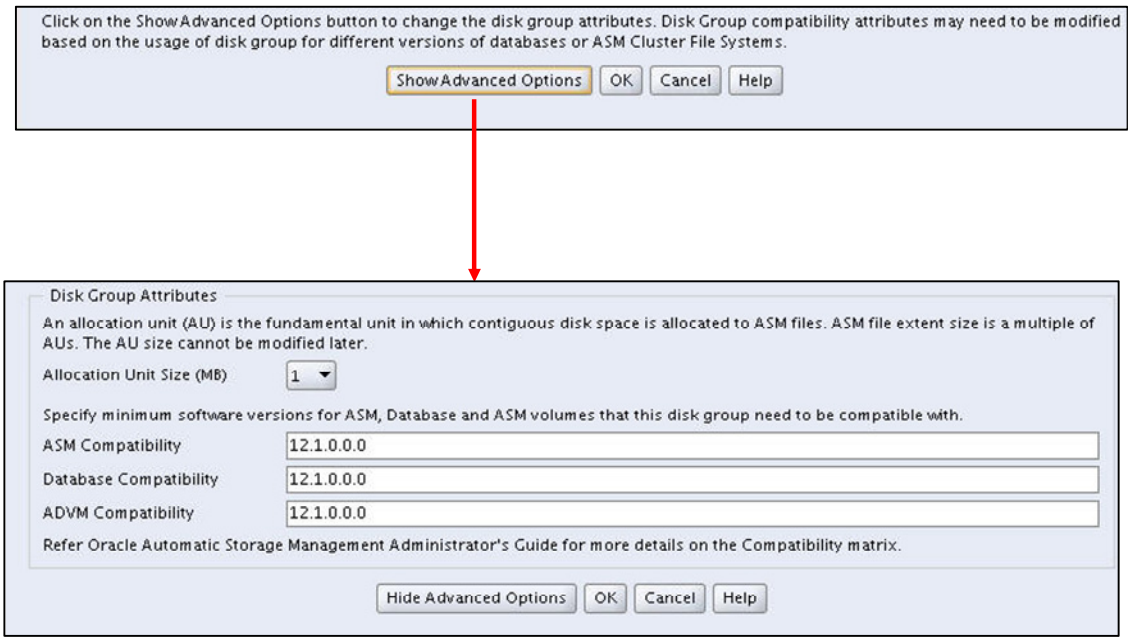





In the slide, the ASMCA utility is being used to create a disk group. Redundancy is set to Normal. Four disks are selected to be members of this disk group: `ORCL:DATA5`, `ORCL:DATA6`, `ORCL:DATA7`, and `ORCL:DATA8`. The first two disks are placed in the failure group ACFS1, and the second two disks are in the failure group ACFS2. The header status can show multiple valid values. When this is the first time 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 disks that are members of existing disk groups. The default string will allow ASM to find disks that have been initialized with the Oracle ASMLib utility, `oracleasm`. Advanced options for disk groups are shown later in this lesson.

# 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. On this portion of the page, you can set disk group attributes: allocation unit size and compatibility parameters.

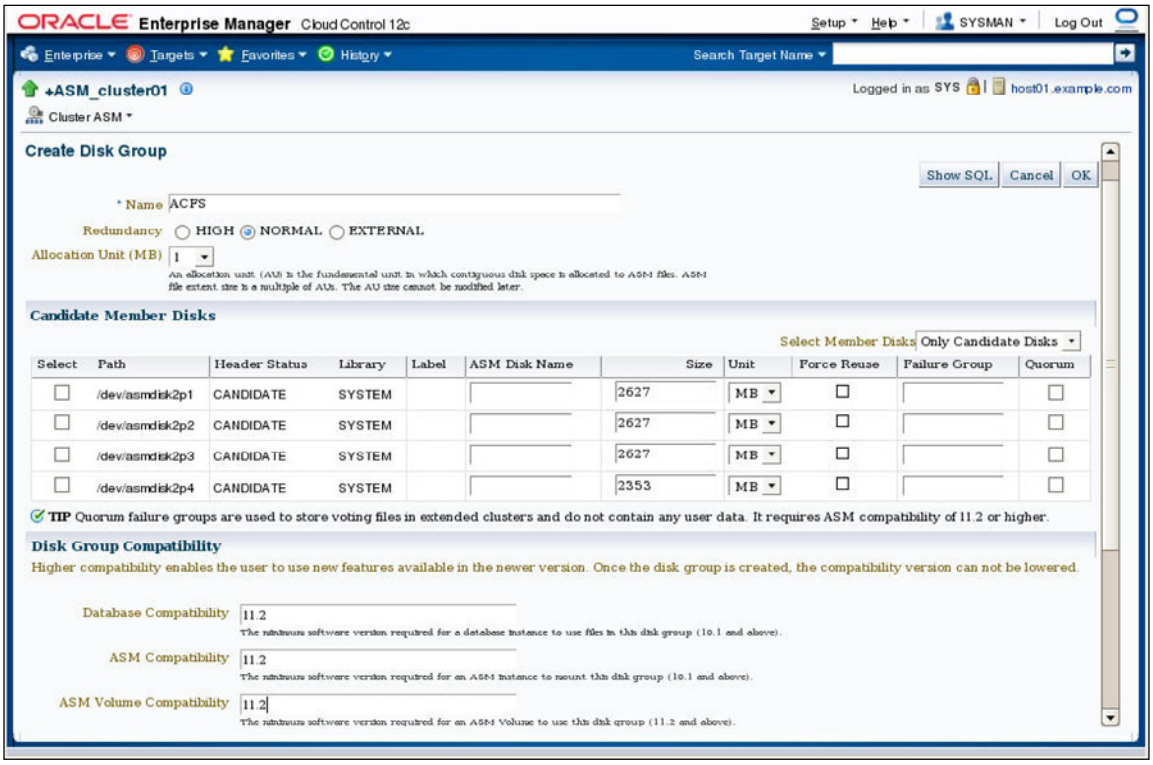
The field labeled ASM Compatibility sets the `COMPATIBLE.ASM` attribute. For Oracle ASM in Oracle Database 12*c*, 12.1 is the default setting for the `COMPATIBLE_ASM` attribute when using the SQL `CREATE DISKGROUP` statement, the ASMCMD `mkdg` command, and ASMCA.. When setting the values for the compatibility 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 “11.2”; Oracle assumes that any missing version number digits are zeroes.

The 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 using the SQL `CREATE DISKGROUP` statement, the ASMCMD `mkdg` command, and ASMCA.

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

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

# Creating a Disk Group with Enterprise Manager






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;
```



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 using database management tools such as the ASM Configuration Assistant (ASMCA), Oracle Enterprise Manager, and 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 for 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, 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, then 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 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, then ASM returns an error. If it points to one or more disks that have already been assigned to a different disk group, then 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 the 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, then that label is used as the disk name. If you are not using ASMLib, then 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, then ASM returns an error. If you specify a size less than the capacity of the disk, you limit the disk space 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 in this way 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.

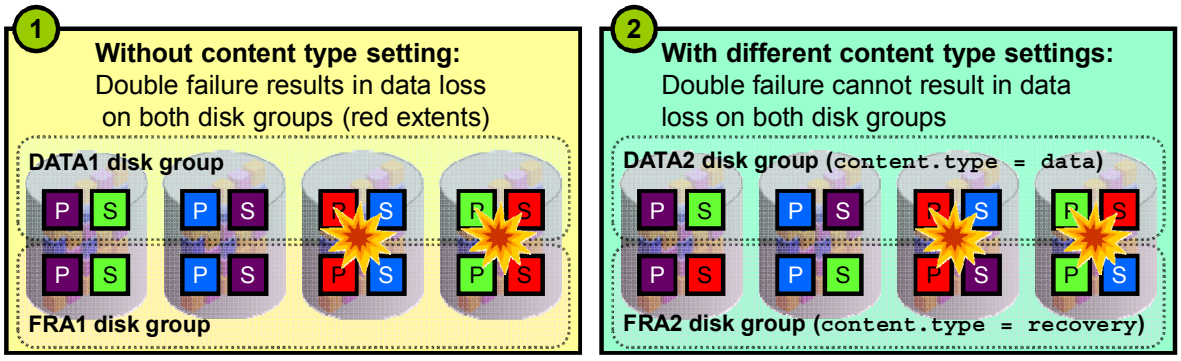
# Specifying Content Type for a Disk Group

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';
```

- Decreases the likelihood that multiple failures 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 prior versions of ASM, the same algorithm and adjacency measure was 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 is 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 one shows two `NORMAL` redundancy disk groups, `DATA1` and `FRA1`, that 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 far right results in data loss in both disk groups; that is, the red extents. In this case, it is possible that the double-failure could result in the loss of a data file and the archived log files required to recover it.

Example two shows `NORMAL` redundancy disk groups, `DATA2` and `FRA2`, 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` is spread differently across the disks.

In this example, failure of the two disks at the far right results in data loss only in the `DATA2` disk group. However because of the different distribution of data that is associated with the different content type setting, `FRA2` experiences no data loss. In this case, the 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.

# Renaming Disks Groups

- The `renamedg` utility enables you to change the name of a cloned disk group.
- `renamedg` renames a disk group 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` 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 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. Allowed values 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, then 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, then 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` using a disk string to locate the disks and the verbose option is enabled. The second example only creates a configuration file during the completion of phase one of the `renamedg` operation. To run phase two of the `renamedg` operation using the configuration file 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

# 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. Attribute can be set only during disk group creation. | 1MB, 2MB, 4MB, 8MB, 16MB, 32MB, 64MB | 1MB |
| 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 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 required for an ASM instance to mount this disk group | At least the first two digits of a valid Oracle Database release number | 12.1 |
| 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 required for a database instance to use files in this disk group, whereas COMPATIBLE.ASM specifies the minimum software version 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. The COMPATIBLE.ASM must be 11.2 or greater 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/disables content checking when performing data copy operations for rebalancing | TRUE or FALSE | FALSE |
| CONTENT.TYPE | This attribute identifies the disk group type. | DATA, SYSTEM, or RECOVERY | DATA |
| IDP.BOUNDARY | Used to configure Oracle Exadata storage. | AUTO | AUTO |
| IDP.TYPE | Used to configure Oracle Exadata storage. | DYNAMIC | DYNAMIC |
| SECTOR_SIZE | Used with CREATE DISKGROUP to specify the disk sector size | 512,4096,4K | Platform dependant |
| 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 only valid for disk groups that are set to normal or high redundancy. `COMPATIBLE.ASM` must be set to 11.2.0.3 or higher.

**IDP.BOUNDARY** and **IDP.TYPE** are used to configure Oracle Exadata storage.

**SECTOR_SIZE** specifies the sector size for disks in a disk group and can only be set when creating a disk group. The values can be set to 512, 4096, or 4K if the disks support those 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.

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.

**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, then all disks in the disk group must be of that type. If the attribute is set to OTHER, then any types of disks can be in the disk group. If the STORAGE.TYPE disk group attribute is set to AXIOM or ZFSSA, then 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 higher.

# 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/disables the functionality to discard unused storage space after a rebalance | TRUE OR FALSE | FALSE |

ORACLE



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 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. The 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 there is a repair time 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 only be set when altering a disk group and is only applicable 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 a more efficient overall physical storage utilization.

# Viewing Disk Group Attributes

- In an ASM instance, V$ASM_ATTRIBUTE displays one row for each attribute defined.
- In addition to attributes specified by CREATE DISKGROUP and ALTER DISKGROUP statements, the view may show other attributes that are created automatically.
- Disk group attributes are listed in V$ASM_ATTRIBUTE only if the disk group attribute COMPATIBLE.ASM is set to 11.1 or higher.
- The same information can be displayed 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 higher. If COMPATIBLE.ASM is not set to 11.1 or higher, then V$ASM_ATTRIBUTE will have no rows corresponding to the disk group.
Following is an example of a query that retrieves attributes for a particular disk group:

```
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               12.1.0.0.0  N   Y
DATA        compatible.asm                12.1.0.0.0  N   Y
DATA        compatible.rdbms              12.1.0.0.0  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 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 | 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 | >=11.1 |
| 12.1 | 12.1 | 12.1 | >=12.1 | >=12.1 |

ORACLE



You can set disk group compatibility with the `CREATE DISKGROUP` or `ALTER DISKGROUP` SQL statement. For example:

```
ALTER DISKGROUP FRA SET ATTRIBUTE 'compatible.asm' = '12.1';
```

In Oracle Database 12*c*, 12.1 is the default setting for the `COMPATIBLE.ASM` attribute when using the `SQL CREATE DISKGROUP` statement, the `asmcmd mkdg` command, and ASMCA.

Before advancing the `COMPATIBLE.RDBMS` attribute, ensure that the values for the `COMPATIBLE` initialization parameter for all of the databases that access the disk group are set to at least the value of the new setting for `COMPATIBLE.RDBMS`. For ASM in Oracle Database 12*c*, 10.1 is the default setting for the `COMPATIBLE.RDBMS` attribute when using the `SQL CREATE DISKGROUP` statement, the `asmcmd mkdg` command, and ASMCA.

When setting the values for the `COMPATIBLE` attributes, specify at least the first two digits of a valid Oracle Database release number. For example, you can specify compatibility as `11.2` or `12.1`. ASM assumes that any missing version number digits are zeroes.

Note that although it is possible to alter these attributes, they can only be advanced (for example, from `10.1` to `11.1`). Advancing the values for on-disk compatibility attributes is an irreversible operation. To revert to the previous value, you must create a new disk group with the old compatibility attributes and then restore the database files that were in the disk group.

# 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 shown in V$ASM_ATTRIBUTE | >=11.1 | >=11.1 | n/a |
| Fast mirror resync (DISK_REPAIR_TIME) | >=11.1 | >=11.1 | n/a |
| Exadata storage V2 | >= 11.1.0.7 | >= 11.1.0.7 | n/a |
| Variable size extents | 11.1 | 11.1 | n/a |
| Intelligent Data Placement | >=11.2 | >=11.2 | n/a |
| OCR and voting disks in a disk group | >=11.2 | n/a | n/a |
| Sector size set to nondefault value | >=11.2 | >=11.2 | n/a |
| Oracle ASM SPFILE in a disk group | >=11.2 | n/a | n/a |
| Oracle ASM File Access Control | >=11.2 | >=11.2 | n/a |
| Volumes in disk groups | >=11.2 | n/a | >=11.2 |
| Read-write snapshots | >= 11.2.0.3 | n/a | >= 11.2.0.3 |
| Replication status of a disk group | >= 12.1 | n/a | n/a |
| Storing shared password file in disk group | >= 12.1 | n/a | >= 12.1 |
| Storing data files and redo logs in ACFS | >= 12.1 | n/a | >= 12.1 |
| Creation from existing snapshot | >= 12.1 | n/a | >= 12.1 |

ORACLE



Setting the disk group compatibility attributes is essentially a balance between flexibility and functionality.

At one extreme setting, `COMPATIBLE.RDBMS` and `COMPATIBLE.ASM` to `10.1` will allow the greatest flexibility regarding the ASM and database software versions that can use a disk group. However, such a setting limits access to various Oracle Database 11*g* ASM features such as large (>8 MB) allocation units, fast mirror resync, variable sized extents, and preferred read failure groups.

That is, setting `COMPATIBLE.RDBMS` and `COMPATIBLE.ASM` to `11.1` will provide access to the Oracle Database 11*g* ASM features but require that the Oracle Database 11*g* software be installed and in use across your environment.

# 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 those values.
  - The default value is platform dependent.
- In native mode, there is no performance penalty with:
  - ASM files, at the disk group level
- `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` attributes must be set to 11.2 or higher 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.



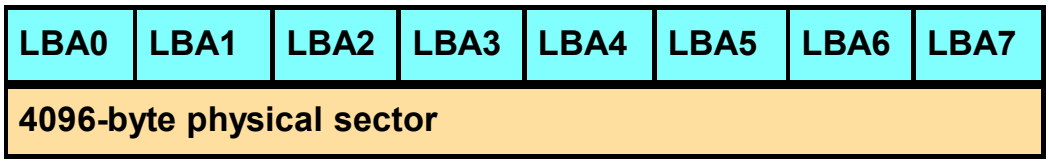
You can use the optional `SECTOR_SIZE` disk group attribute with the `CREATE DISKGROUP` SQL statement to specify disks with the sector size set to the value of `SECTOR_SIZE` for the disk group. Oracle ASM provides support for 4 KB sector disk drives without negatively affecting performance. The `SECTOR_SIZE`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

The values for `SECTOR_SIZE` can be set to 512, 4096, or 4K if the disks support those 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.

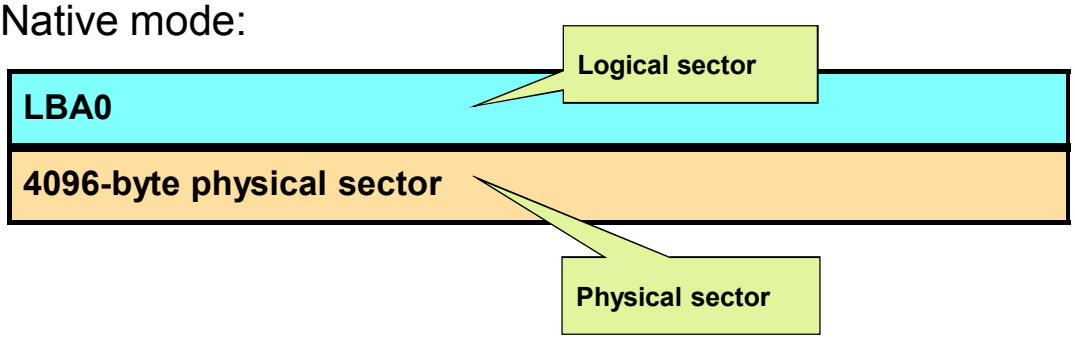
Oracle ACFS does not support 4 KB sector drives directly. There is a performance penalty for Oracle ACFS when using 4 KB sector disk drives in 512 sector emulation mode.

# Supporting 4 KB Sector Disks

- Emulation mode:




- Native mode:






4 KB sector disks have physical sectors (shown in gray) and logical sectors (shown in 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 hurt 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 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
- All the disks in a disk group must have the same sector size.
- Restrictions on redo log files:
  - The `BLOCKSIZE` redo log file and the `SECTOR_SIZE` disk group



ASM sector size can be managed from 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, the 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 the 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 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, then 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 disks discovered through disk search strings have a sector size 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, then 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 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, then 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 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'='11.2','compatible.rdbms'='11.2',
    'sector_size'='4096';
```



Use the optional SECTOR_SIZE clause in the CREATE DISKGROUP command to explicitly specify the 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. The values for SECTOR_SIZE can be set to 512, 4096, or 4K. ASM provides support for 4 KB sector disk drives without negatively affecting performance. The example below shows the contents of an XML file used to create a disk group 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 a 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.

# 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 available to the database instance.
- Other ASM dynamic performance views relate to V$ASM_DISKGROUP through the GROUP_NUMBER column.
- ASMCMD lsdg command provides the same information.
- V$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.



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 the 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, that means 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 for mirroring. It is supposed to show the amount of free space that can be safely utilized taking mirroring into account and yet be able 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.





Following is an example of a query that retrieves information about disks in a specific disk group:

```
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
```

Following is another example that uses V$ASM_DISK to display disks that do not belong to any disk group:

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

- Adding 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 the 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 have been 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 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
```



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 will fail 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/rebalance operation 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 a 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 the data to another disk is effectively disabled, the disk will remain 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 the data to another disk is effectively disabled, the disk will remain in this state indefinitely until another add or drop operation causes the disk group to be rebalanced or the disk group is manually rebalanced using the `ALTER DISKGROUP <diskgroup_name> REBALANCE` statement.

# V$ASM_OPERATION

- In an ASM instance, V$ASM_OPERATION displays one row for every active long-running operation executing in the ASM instance.
- In a database instance, V$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
```



V$ASM_OPERATION provides information about long-running operations conducted 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 listing all the 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 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
```

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

ORACLE



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 disks in a disk group 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 only use the `RENAME DISK` operation when the disk group that contains the disk is in the `MOUNT RESTRICTED` state. If any disks in the disk group are offline, then the `RENAME` operation fails. If the new disk name exists, then 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 disks in the disk group
  - Resize specific disks
  - Resize all of the disks in a specified failure group
- If the size of the disk is increasing, then 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 of the disks in failure group `failgrp1` 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 disks in the disk group
- Resize specific disks
- Resize all of the disks in a specified failure group

If you do not specify a new size in the `SIZE` clause, then 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, then 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, then the new size is made permanent, otherwise the rebalance fails.

The example above resizes all of the disks in failure group `failgrp1` of disk group `data1`. 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 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 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 will fail, 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 will fail.

ASMCMD can be used to mount and mount disk groups as shown above. 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

To view ASM 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
```



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

```
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 an disk group, including contents, if required:

```
ASMCMD> dropdg DATA2;
```

```
ASMCMD> dropdg -r DATA2;
```

```
ASMCMD> dropdg -r -f DATA3;
```

ORACLE



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.

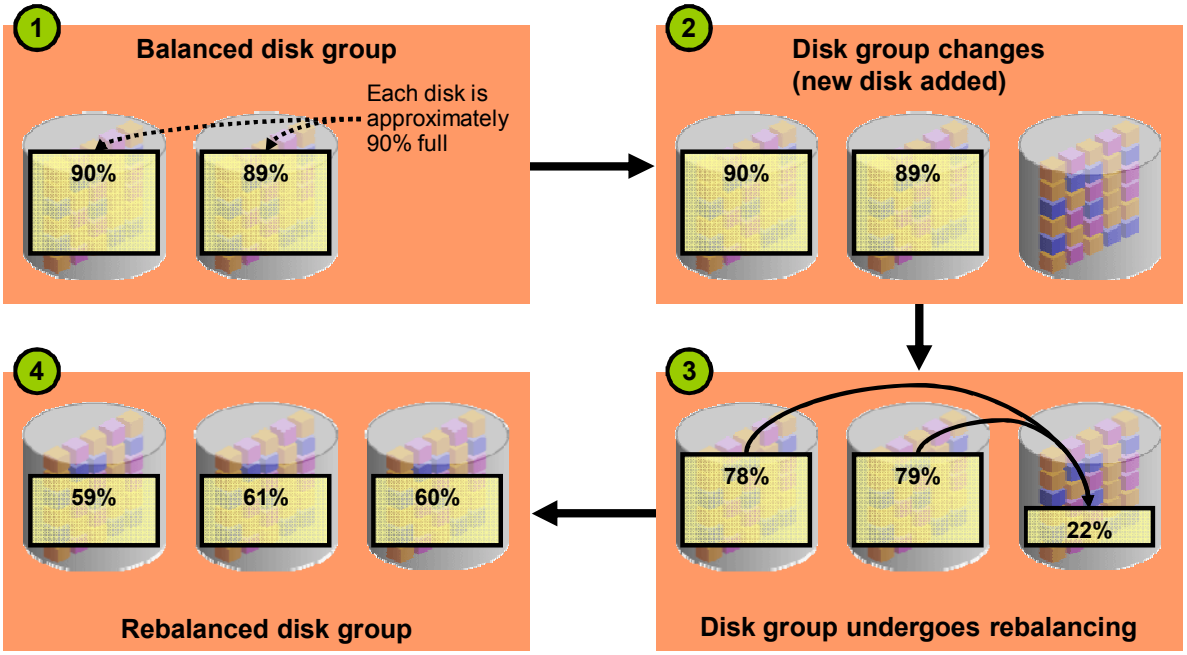
In order for the `DROP DISKGROUP` statement to succeed, the disk group must be mounted by the current ASM instance only and have none of the disk group files open.

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 groups and it's contents. Use the `-f` option to force the operation.

# ASM Disk Group Rebalance: Review






An ASM disk group is balanced when there is an even distribution of data across all of the available disks. In the 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. Enhancement have been made to ASM disk group rebalance 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 previous 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.



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 the 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 then 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
```

ORACLE



Previous ASM releases provided work estimates for ASM rebalance operations. 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, then 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 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 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 higher, the rebalance operation may be run as one process using asynch 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, then 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$ASM_ESTIMATE view. The EXPLAIN WORK SQL statement determines the amount of work for a rebalance operation and the resulting calculations are displayed in the V$ASM_ESTIMATE view as shown in the slide above. The EST_WORK column provides an estimate of the number of allocation units that have to be moved by the rebalance operation to complete.

The PASS column of V$ASM_OPERATION is updated for 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
```

# Proactively Validating Data Integrity

In previous versions, data is checked for logical consistency when it is read.

- If a logical corruption is detected:
  - Automatic recovery can be performed by using the mirror copies.
  - Manual recovery can also be performed 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.

ORACLE



In previous 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 also 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 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';
```



Previously, 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 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 on the slide. The progress of on-demand scrubbing operations can be monitored by using the `V$ASM_OPERATION` view.

Following are details regarding 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 above 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 about the corruption and the reason of the failed repair attempt are written into 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 associated with scrubbing.

# Checking the Consistency of Disk Group Metadata

You can check the internal consistency of disk group metadata 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 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 about 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.

# Managing Capacity in Disk Groups

- In a NORMAL or HIGH redundancy disk group, there must be enough capacity to manage re-creation of data lost after a failure of one or two failure groups.
  - If there is not enough space, then 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.
- The `REDUNDANCY_LOWERED` column in `V$ASM_FILE` provides information about files with reduced redundancy.
- 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.

ORACLE



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 a re-creation of data that is lost after a 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, then 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. The `REDUNDANCY_LOWERED` column in the `V$ASM_FILE` view provides information about files with reduced redundancy. 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 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/
```



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, this worst failure refers to a permanent failure where the disks must be dropped, not the case where the disks go offline and then 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 equal to twice their size, if 4 GB of actual usable file space remains, then USABLE_FILE_MB equals roughly 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. The 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, then 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 because of the imbalance in the disk group. Ensure that you initiate a manual rebalance to force even data distribution which results in an accurate presentation of the values in the FREE_MB column.

  With fine grain striping using 128 KB, the storage is preallocated 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 preallocation.

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

The results from the following query show capacity metrics for a normal redundancy disk group that consists of six 1 GB (1024 MB) 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
```

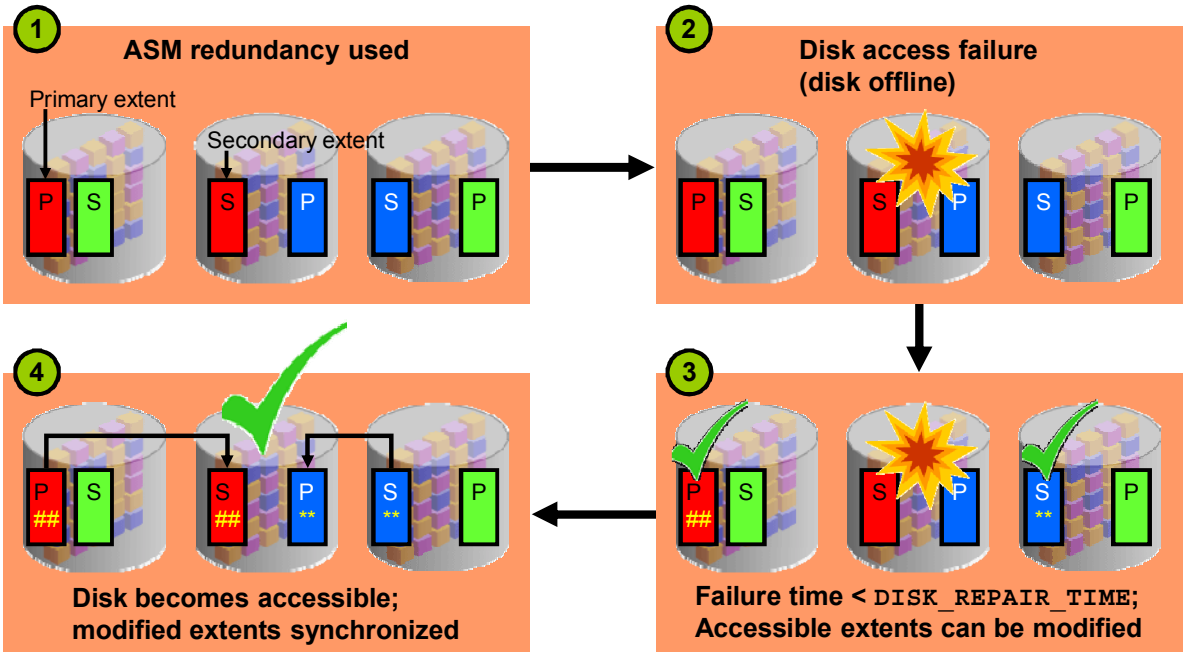
The REQUIRED_MIRROR_FREE_MB column shows that 1 GB of extra capacity must be available to restore full redundancy after one or more disks fail. The first three numeric columns in the query results are raw numbers. That is, they do not take redundancy into account. Only the last column is adjusted for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

# ASM Fast Mirror Resync: Review

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 have been 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.

Numerous enhancement have been made to ASM fast mirror resync in Oracle Database 12*c*. The following pages introduce those enhancements.

# 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.
- Range is `1` (least resources) to `1024` (most resources).
- If 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 previous versions, which allowed administrators to control the amount of resources that are 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 previous versions:
  - Administrators had to manually reexecute the affected command.
  - The entire resync operation had to be reexecuted.
- 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 prior to the interruption.

ORACLE



A disk online operation, and 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 has to manually reissue the interrupted command. Also, the metadata used to identify the extents that require resyncing (the stale extents) was only cleared at the end of the resync operation. If the resync operation was interrupted for any reason, the entire operation must be reexecuted.

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 now 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 slide shows an example of the new time estimates that accompany resync operations displayed in the V$ASM_OPERATION view.

# Even Read

In previous versions:

- By default, ASM always reads the primary copy of mirrored data if it is available.
- Alternatively, preferred read failure groups could be configured.

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.



In previous versions, the default behavior for ASM is to always read the primary copy of a mirrored extent unless a failure condition requires otherwise.

Alternatively, administrators can configure preferred read failure groups, by using the `ASM_PREFERRED_READ_FAILURE_GROUPS` instance parameter, to specify the failure group from which each ASM instance should read.

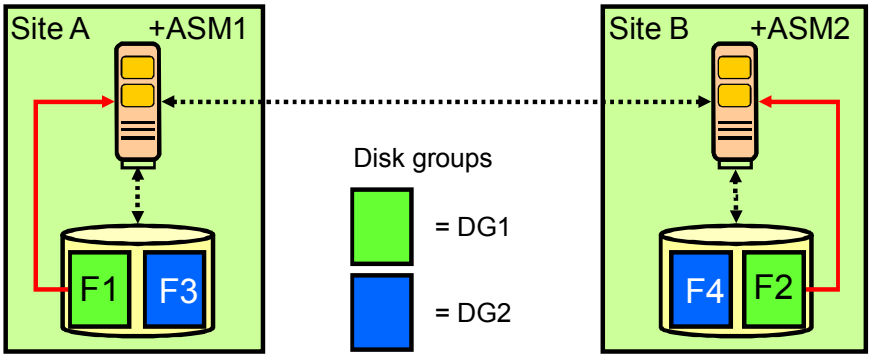
The Even Read feature of ASM, introduced in release 12.1, distributes data reads evenly across all disks in a disk group. For each I/O request presented to the system, one or more disks may contain the data. With Even Read enabled, each read request is sent to the least loaded of the available disks.

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.

Note that Even Read only attempts to equalize the number of reads on each disk across each disk group. It does not measure the latency or performance of each read. Therefore, Even Read does not replace the functionality that is provided by preferred read failure groups.

# Preferred Read Failure Groups

In a multisite cluster, with separate storage at each site, preferred read failure groups allow every node in the cluster to read from its local disks resulting in better performance.




```
+ASM1.ASM_PREFERRED_READ_FAILURE_GROUPS = DG1.F1, DG2.F3
+ASM2.ASM_PREFERRED_READ_FAILURE_GROUPS = DG1.F2, DG2.F4
```

ORACLE



Before Oracle Database 11*g*, ASM always read the primary copy of a mirrored extent. It may be more efficient for a node to read from a failure group extent that is closest to the node, even if it is a secondary extent. This is especially true in extended (multisite) cluster configurations where reading from a local copy of an extent provides improved performance.

Now you can do this by configuring the preferred mirror read using the new initialization parameter, `ASM_PREFERRED_READ_FAILURE_GROUPS`. The disks in the identified failure groups become the preferred read disks. Thus, every node can be configured to read from its local extent copy. This results in higher efficiency and performance, and reduced network traffic. The setting for this parameter is instance specific.

If there is more than one mirrored copy and you have set a value for the `ASM_PREFERRED_READ_FAILURE_GROUPS` parameter, ASM first reads the copy that resides on a preferred read disk. If that read fails, ASM attempts to read from another mirrored copy that might not be on a local preferred read disk.

You can use the `PREFERRED_READ` column in the `V$ASM_DISK` view to determine whether a particular disk belongs to a preferred read failure group.

To identify performance issues with the ASM preferred read failure groups, use the `V$ASM_DISK_IOSTAT` view. This view displays input/output statistics for each ASM client.

# 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';
```

ORACLE



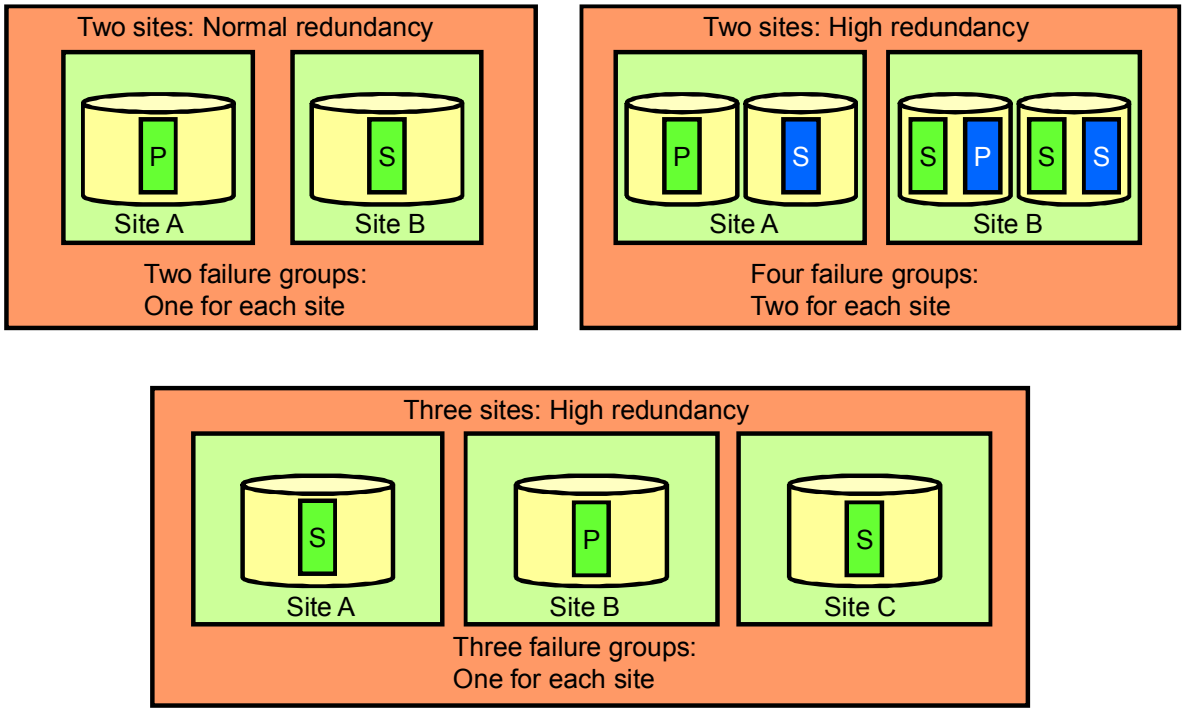
When individual disks fail, the failure is often terminal and the disk must be replaced. When all the disks in a failure group fail simultaneously, it is unlikely that all the disks individually failed at the same time. Rather it is more likely that some transient issue caused the failure. For example, 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 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 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 using a fast mirror resync. If the failure is not corrected in time, ASM must automatically drop all disks in the failure group.

# 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. In this way, ASM can guarantee that each extent has a mirror copy 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 just the disk statistics.
- ASMCMD iostat shows a subset of the statistics depending on the option.



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 just 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 fast recovery area.
- 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.



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.
- 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 the 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 disks in a disk group have similar performance characteristics. For example, the newest and fastest disks might reside in a disk group reserved for the database work area, and slower drives could reside in a disk group 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

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 will fail. 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, then rebalancing will be 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
- Explain key performance and scalability considerations for ASM disk groups

# Practice 4 Overview: Administering ASM Disk Groups

This practice covers the following topics:

- Configuring disk groups
- Adding and removing disks
- Controlling rebalance operations
- Monitoring disk and disk group I/O statistics

5

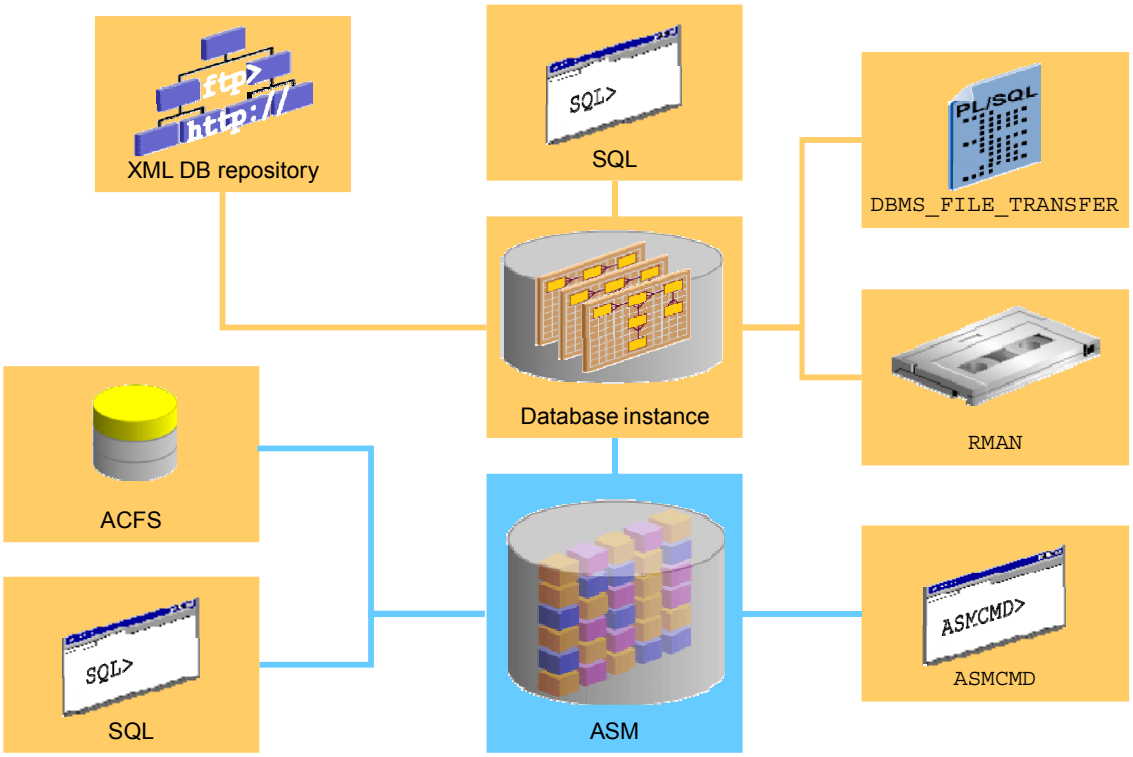
# Administering ASM Files, Directories, and Templates

# 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

# ASM Clients





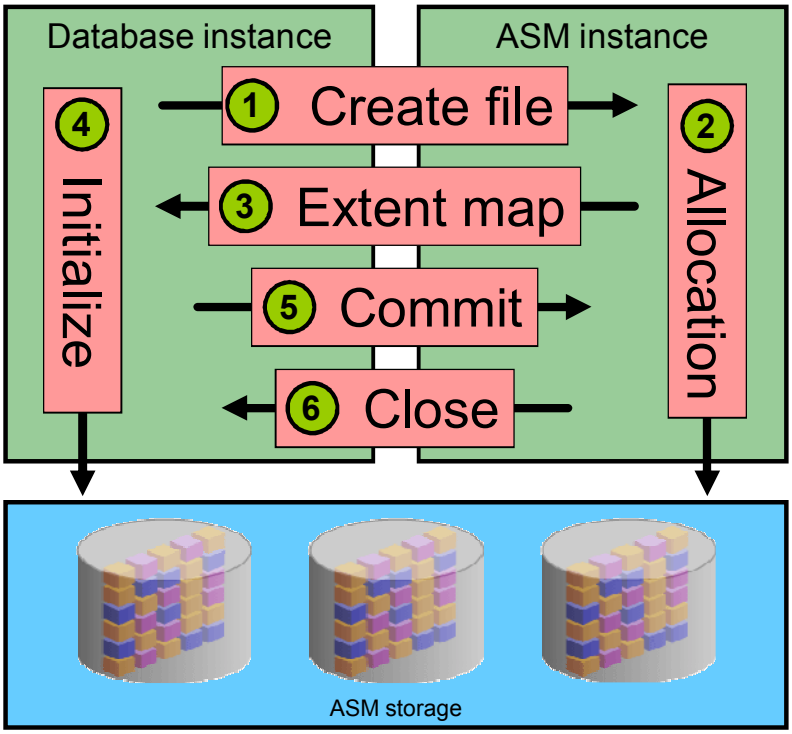

Oracle ASM is a volume manager and a file system for Oracle database files. This slide introduces various ASM clients. Many of these will be examined in greater detail later in this lesson. Clients that connect directly to ASM include:

- **Oracle Database:** Oracle Database is the most fundamental ASM client. It makes requests of 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 the Oracle Cluster Registry (OCR), voting files, or 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:** 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:** 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 various 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`) will implicitly interact with the underlying ASM instance. A database instance will also directly read and write 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) will 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 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 the 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.




The 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 the 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 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 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 will be 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.

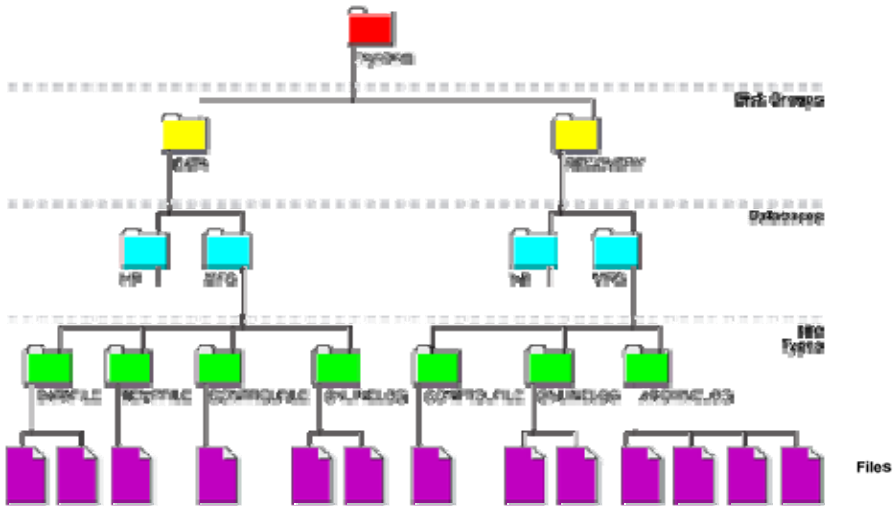
```
RESTORE CONTROLFILE FROM '/u01/c1.ctl';
ALTER DATABASE MOUNT;
BACKUP AS COPY DATABASE FORMAT '+YOURDG';
SWITCH DATABASE TO COPY;
# Repeat command for all online redo log members
SQL "ALTER DATABASE RENAME '/u01/log1' TO '+YOURDG' ";
ALTER DATABASE OPEN RESETLOGS;
# Repeat commands for all temporary 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 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 /
```



The DBMS_FILE_TRANSFER package provides procedures to copy ASM files within a database or transfer binary files between 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> 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> cd ..
ASMCMD> pwd
+DATA/orcl/DATAFILE
ASMCMD> exit
[grid@host01 ~]$
```



ASMCMD is a command-line utility that you can use to view and manipulate 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 `$ORACLE_HOME`, the utility attempts to connect `AS SYSASM`. The ASMCMD utility attempts to connect `AS SYSASM`, so the user that invokes ASMCMD must be a member of the OSASM group. When connecting from a `$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 created in ASM gets a system-generated file name, 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>/<dbname>/<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
```




Every file created in ASM gets a system-generated file name, 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 displaying Oracle Database file names, such as `V$DATAFILE` and `V$LOGFILE`.

A fully qualified file name has the following form:

`+group/dbname/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.
- `dbname` is the `DB_UNIQUE_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 dump set, 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 choosing. Alias file names are implemented using a hierarchical directory structure, with the slash (/) or backslash (\) character separating name components. You can create an alias in any system-generated or user-created ASM directory already in existence except for 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 created 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 just 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 ASM disks in the `DATA` disk group. File extents are stored 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 |



The table in the slide specifies the valid contexts for each file name form.

A 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`.

A 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, the SIZE defaults to 100 MB and the file is autoextensible 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
```



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 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`.

# View ASM Aliases, Files, and Directories

- In an ASM instance, V$ASM_ALIAS displays one row for each alias, system-generated file name, and directory present in every disk group mounted by the ASM instance.
- V$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 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 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

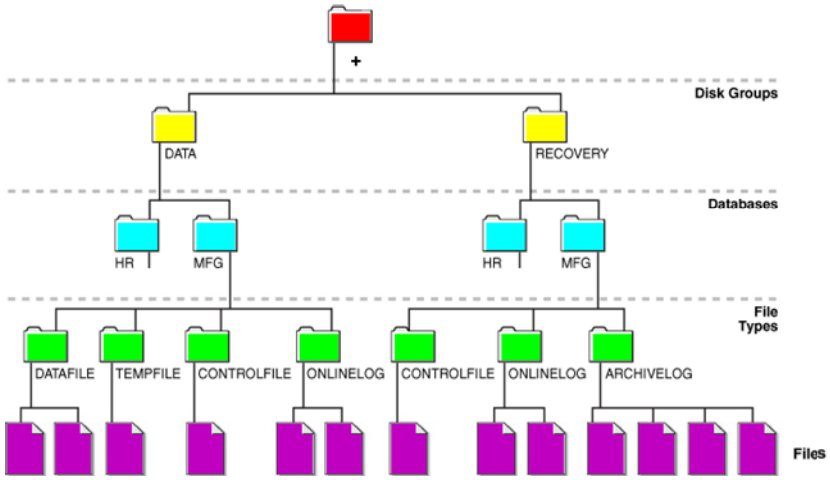


V$ASM_FILE contains information about the physical attributes of ASM files. This includes settings for striping, redundancy, and block size, along with information about the file size, type, redundancy status, creation date, and date of last modification.

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 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 files in the `DATA` disk group. The `orcl` directory is the parent directory for all files in the `orcl` database, and the `DATAFILE` directory contains all data files for the `orcl` database.

You can create your own directories within this hierarchy to store 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';
```

ORACLE



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 containing alias names unless you also specify the `FORCE` clause. If you specify `DROP DIRECTORY ... FORCE`, ASM will recursively remove 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:

- `mkdir`: Makes 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';
```



ASM alias file names, or aliases, provide a more user-friendly means of 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 created in this manner 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 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, redundancy, and region attributes for ASM files.
- Striping attribute options are:
  - `FINE`: 128 KB stripe size
  - `COARSE`: 1 AU stripe size
- 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.



Templates are used to set redundancy (mirroring) and striping attributes of each individual file 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 written to that disk group. 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 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. The region values for all the system templates are set to `COLD` and `MIRRORCOLD`.

| 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 either an ASM or database instance, V$ASM_TEMPLATE displays one row for every template present in every disk group 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
...
```



The V$ASM_TEMPLATE view displays disk group templates and their attribute settings. The following query lists all the templates associated with the DATA disk group. The SYSTEM column returns Y for system-generated templates and N for user-defined templates.

```
SQL> SELECT T.NAME, 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 will default 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 will only be applied to new files that reference the modified template, either explicitly or implicitly. You can only drop user-defined templates, not system-default templates.

The `V$ASM_TEMPLATE` view lists all of 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;
```



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 on 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 will apply the named template to files created by statements such as `CREATE TABLESPACE` and `ALTER TABLESPACE`.

# Intelligent Data Placement

- Intelligent Data Placement enables you to specify disk regions on Oracle ASM disks for best performance.
- Frequently accessed data can be placed on the outermost (hot) tracks which have greater speed and bandwidth.
- Intelligent Data Placement works best for the following:
  - Databases with data files that are accessed at different rates.
  - Disk groups that are more than 25% full.
  - Disks that have better performance at the beginning of the media relative to the end.
- Intelligent Data Placement leverages the geometry of the disk, so it is well suited to JBOD (just a bunch of disks).
- In contrast, a storage array with LUNs composed of concatenated volumes masks the geometry from ASM.

ORACLE



Intelligent Data Placement enables you to specify disk regions on Oracle ASM disks for best performance. Using the disk region settings, you can ensure that frequently accessed data is placed on the outermost (hot) tracks which have greater speed and higher bandwidth. In addition, files with similar access patterns are located physically close, reducing latency. Intelligent Data Placement also enables the placement of primary and mirror extents into different hot or cold regions.

Intelligent Data Placement settings can be specified for a file or in disk group templates. The disk region settings can be modified after the disk group has been created. The disk region setting can improve I/O performance by placing more frequently accessed data in regions furthest from the spindle, while reducing your cost by increasing the usable space on a disk.

Intelligent Data Placement works best for the following:

- Databases with data files that are accessed at different rates. A database that accesses all data files in the same way is unlikely to benefit from Intelligent Data Placement.
- Disk groups that are more than 25% full. If the disk group is only 25% full, the management overhead is unlikely to be worth any benefit.
- Disks that have better performance at the beginning of the media relative to the end. Because Intelligent Data Placement leverages the geometry of the disk, it is well suited to JBOD (just a bunch of disks). In contrast, a storage array with LUNs composed of concatenated volumes masks the geometry from Oracle ASM.

# Enabling Intelligent Data Placement

- The COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes must be set to 11.2 or higher.
- Can be managed with the ALTER DISKGROUP ADD, MODIFY TEMPLATE and MODIFY FILE SQL statement:

```
ALTER DISKGROUP data ADD TEMPLATE datafile_hot
ATTRIBUTE (
HOT
MIRRORHOT);
```

*Or*

```
ALTER DISKGROUP data MODIFY FILE
'+data/orcl/datafile/users.259.679156903'
ATTRIBUTE (
HOT
MIRRORHOT);
```

ORACLE



The COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes must be set to 11.2 or higher to use Intelligent Data Placement. Intelligent Data Placement can be managed with the ALTER DISKGROUP ADD or MODIFY TEMPLATE SQL statements and the ALTER DISKGROUP MODIFY FILE SQL statement.

- The ALTER DISKGROUP TEMPLATE SQL statement includes a disk region clause for setting hot/mirrorhot or cold/mirrorcold regions in a template:

```
ALTER DISKGROUP data ADD TEMPLATE datafile_hot
ATTRIBUTE (
HOT
MIRRORHOT);
```

- The ALTER DISKGROUP MODIFY FILE SQL statement that sets disk region attributes for hot/mirrorhot or cold/mirrorcold regions:

```
ALTER DISKGROUP data MODIFY FILE
   '+data/orcl/datafile/users.259.679156903'
ATTRIBUTE (
HOT
MIRRORHOT);
```

When you modify the disk region settings for a file, this action applies to new extensions of the file, but existing file contents are not affected until a rebalance operation. To apply the new Intelligent Data Placement policy for existing file contents, manually initiate a rebalance.

# Viewing Disk Region Information

- Information about Intelligent Data Placement is displayed in the columns of the following views:
  - V$ASM_DISK
  - V$ASM_FILE
  - V$ASM_DISK_IOSTAT
  - V$ASM_TEMPLATE

```
SELECT dg.name AS diskgroup, f.file_number, f.primary_region, f.mirror_region,
f.hot_reads, f.hot_writes, f.cold_reads, f.cold_writes
FROM V$ASM_DISKGROUP dg, V$ASM_FILE f WHERE dg.group_number = f.group_number and
dg.name = 'DATA';

DISKGROUP FILE_NUMBER PRIM MIRR HOT_READS HOT_WRITES COLD_READS COLD_WRITES
--------- ----------- ---- ---- --------- ---------- ---------- -----------
DATA              257 COLD COLD         0          0     119770      886575
DATA              258 COLD COLD         0          0       1396      222282
DATA              259 COLD COLD         0          0       2056         199
...
```



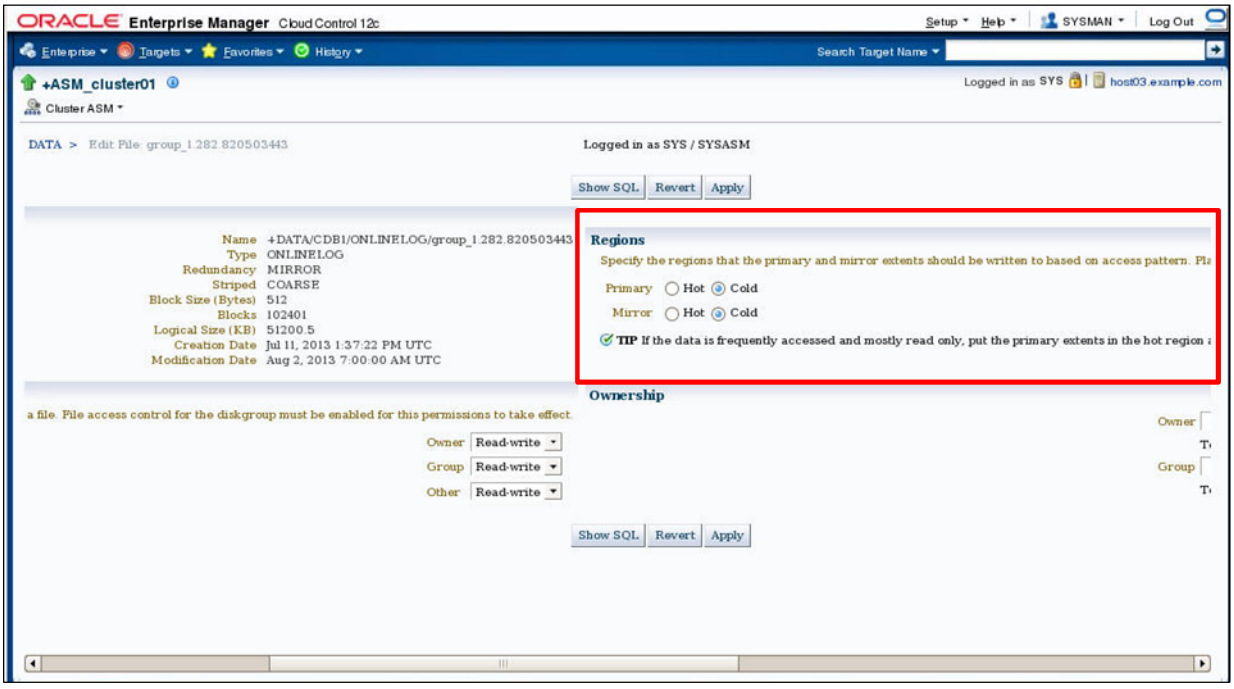
Information about Intelligent Data Placement is displayed in the columns of the V$ASM_DISK, V$ASM_DISK_IOSTAT, V$ASM_FILE, and V$ASM_TEMPLATE views.

The example above retrieves Intelligent Data placement information using V$ASM_FILE. The example below uses V$ASM_TEMPLATE:

```
SELECT dg.name AS diskgroup, t.name, t.stripe, t.redundancy,
t.primary_region, t.mirror_region FROM V$ASM_DISKGROUP dg,
V$ASM_TEMPLATE t WHERE dg.group_number = t.group_number and dg.name
= 'DATA' ORDER BY t.name;

DISKGROUP NAME                 STRIPE REDUND PRIM MIRR
--------- -------------------- ------ ------ ---- ----
DATA      ARCHIVELOG           COARSE MIRROR COLD COLD
DATA      ASMPARAMETERFILE     COARSE MIRROR COLD COLD
DATA      AUDIT_SPILLFILES     COARSE MIRROR COLD COLD
DATA      AUTOBACKUP           COARSE MIRROR COLD COLD
DATA      AUTOLOGIN_KEY_STORE  COARSE MIRROR COLD COLD
DATA      BACKUPSET            COARSE MIRROR COLD COLD
DATA      CHANGETRACKING       COARSE MIRROR COLD COLD
DATA      CONTROLFILE          FINE   HIGH   COLD COLD
...
```

# Assigning Files to Disk Regions with Enterprise Manager






An ASM file can be assigned to a disk region through Enterprise Manager. On the Edit File page, you can assign the primary and mirror extents of a file to either the hot or cold region. An online log file—an example is shown in the slide—with a high volume of writes should be assigned to the hot region for both the primary and mirror extents. The default settings of cold region are shown in the slide's example.

# 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 an 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 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 operating system 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 using 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 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 previous 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



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 used to consolidate many Oracle databases, and can be used to ensure that database administrators can access only the database files that are associated with the databases under their control.

In previous 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 higher
  - `COMPATIBLE.RDBMS` to 11.2 or higher
  - `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. The `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` must be set to 11.2 or higher. 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 write privilege; 6 removes read-write privilege. Concatenating the values gives the 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
```



Access control lists in ASM can be managed with the SQL commands shown.

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 permissions of an Oracle ASM file. Note that setting the 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 the permissions on an open file.

You cannot change the 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
```



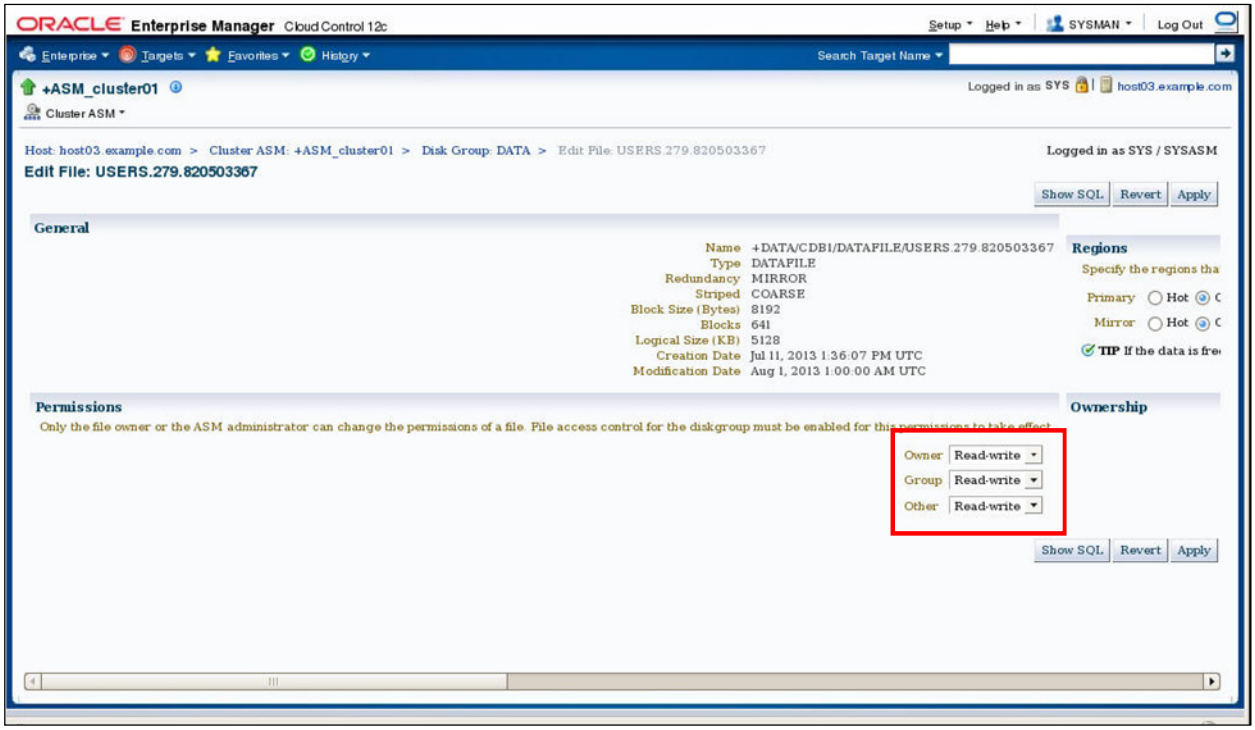
The `ASMCMD` commands allow the storage administrator to perform the same operations as were shown with SQL commands but in a more familiar format. The `asmcmd` commands are patterned after the UNIX commands. The `asmcmd` commands have the same restrictions as the SQL commands.

- `chmod`: Modifies permissions of an Oracle ASM file or list of files. Note that setting the read-only permission to a file that has the 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 the permissions on an open file.
- `chown`: You cannot change the ownership on an open file.

In ASMCMD, the user can connect using the `–a` option to set privileges, either `SYSASM` or `SYSDBA`, but the OS user invoking the command must be a member of the OS group having 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, permissions and ownership of an ASM file can be modified. The 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.



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 clustered.

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 `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 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 of the following 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



**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 5 Overview: Administering ASM Files, Directories, and Templates

This practice covers navigating the ASM file hierarchy, managing aliases, managing templates, and moving files to different disk regions.

# 6

# Administering Oracle CloudFS

# 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 Cloud File System

- Oracle CloudFS components include:
  - **ASM Cluster File System**
  - **Oracle ASM Dynamic Volume Manager**
- Oracle CloudFS can help organizations deploy their applications, databases, and storage in private clouds.
- In previous versions, ACFS provided limited support for Oracle Database files.
- With Oracle Database 12*c*, Cloud FS is supported for storing all Oracle Database files.
  - Entire running databases can be stored inside Cloud FS.
  - Database administrators can leverage Cloud FS services like snapshots, security, and tagging



Oracle Cloud File System (Oracle CloudFS) is designed to help organizations deploy their applications, databases, and storage in private clouds. It delivers a cloud infrastructure that provides network access, rapid elasticity, and provisioning for pooled storage resources that are the key requirements for cloud computing. Customers can use Oracle CloudFS to manage and store all database file types, including general purpose files.

Oracle CloudFS includes Oracle Automatic Storage Management Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM). New Features in Oracle Cloud File System, release 12.1 include:

- High availability NFS
- Support for all Oracle Database files
- Snapshot enhancements
- Advanced auditing
- Metrics plug-in
- Replication enhancements
- Tagging API
- Resource enhancements

# Enhanced Platform Support for Cloud FS Data Services

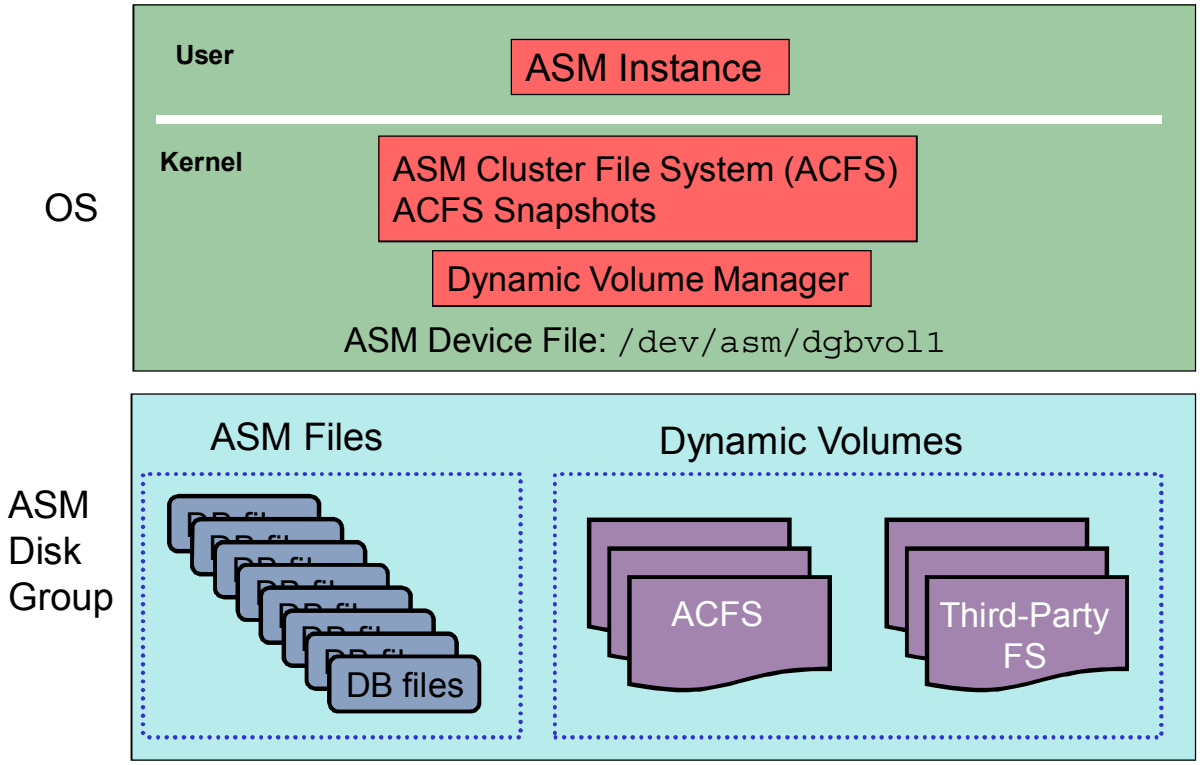
| | Snapshot | Replication | Tagging | Security | Encyption |
|---|---|---|---|---|---|
| **11.2.0.1** | **Read-only** on Linux and Windows | - | - | - | - |
| **11.2.0.2** | **Read-only** on Linux, Wndows, Solaris, AIX | Linux | Linux | Linux | Linux |
| **11.2.0.3** | **Read-write** on Linux, Windows, Solaris, AIX | Linux, Windows | Linux, Windows | Linux, Windows | Linux, Windows |
| **12c** | **Read-write** on Linux, Windows, Solaris, AIX | Linux, Windows, **Solaris, AIX** | Linux, Windows, **Solaris, AIX** | Linux, Windows, **Solaris** | Linux, Windows, **Solaris** |

ORACLE



The table in the slide summarizes the platform support for the different Cloud FS data service capabilities. The platform-specific enhancements for Oracle Database 12*c* are highlighted. No new functionality is introduced; existing features are available on a wider selection of platforms.

# ASM Files and Volumes





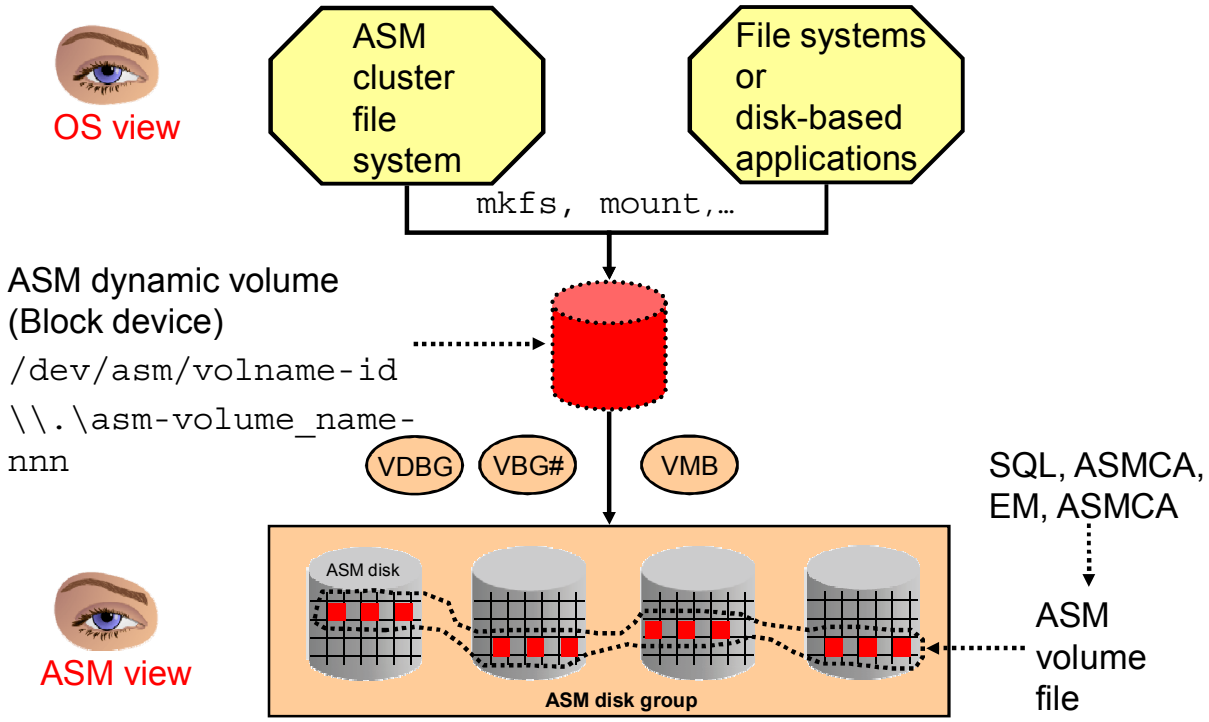

Oracle ASM includes support for a general-purpose cluster file system, the ASM Cluster File System (ACFS). At the operating system (OS) level, the ASM instance provides the disk group, which is a logical container for physical disk space. The disk group can hold ASM database files and ASM dynamic volume 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 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 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 ADMV volume 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 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 accessed through ADVM. File systems or user processes can do 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 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 upon ASM instance startup. The volume device file names are unique clusterwide and persistent across all nodes in the cluster that have an ASM instance running with the disk group mounted and volumes enabled.

Upon Linux system startup, the Oracle clusterware startup will load 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 will then mount any ACFS file systems in the Oracle Cluster Registry (OCR).

Similar actions are performed on Windows.

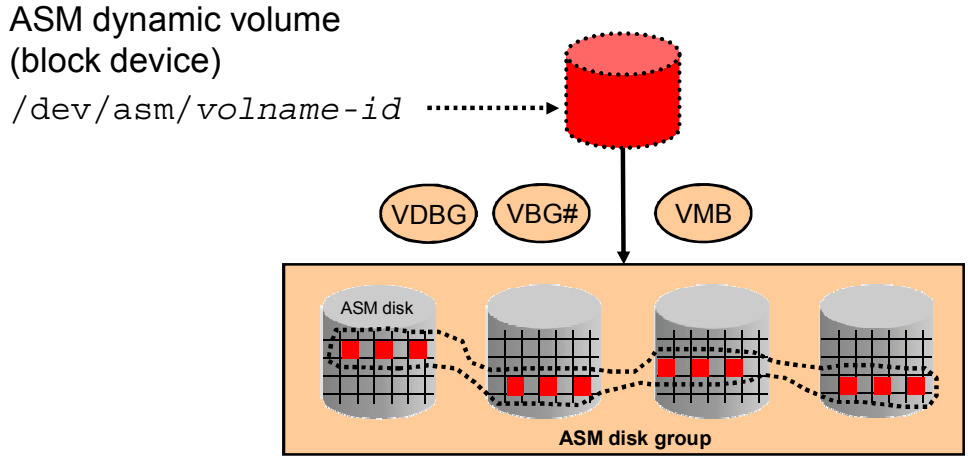
ACFS file systems are accessed through OS file system tools and APIs on UNIX and Linux systems, and accessed through Windows file system tools and APIs on Windows systems. Remote access is supported using standard NAS file access protocols such as network file systems (NFS) and common Internet file system (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/Linux and Administrator privileges on Windows. So, the configuration and loading of the ACFS drivers is performed by the `root` scripts 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 will recover any outstanding metadata transactions on behalf of the failed member. In addition, any lock tokens held by the failed cluster members will be recovered and the failed member will be 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)






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, resize the volume, offline the disk, add or drop a disk, and force and dismount a disk group to the dynamic volume manager driver. The VDBG is a fatal background process so the termination of this process brings down the ASM instance.
- 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 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). The unplanned death of any of these processes does not have an effect on the ASM instance.
- Volume Membership Background (VMB) process provides 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 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 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 is achieved with ACFS metadata checksums and journaling.



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 using native operating system file system application programming interfaces 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 is achieved with ACFS metadata checksums and journaling. ACFS is designed as a multi-node, shared file system model that delivers coherent, cached, direct storage paths to ACFS file data from each cluster member. Oracle ACFS files 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 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 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 the 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.
  - Stripe column default is 4.



ADVM allocates space from an ASM disk group for a volume. The volume allocation unit is the smallest allocation when a volume is created and 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 that are 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 will be 256 MB. When the volume is created, multiples of the VAU will be allocated. If you asked for a volume of 300 MB in size, you would get a volume of 512 MB, assuming 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:

- Stripe column – the number of stripes (default 4)
- Stripe width – the size of each stripe (default 128 KB)



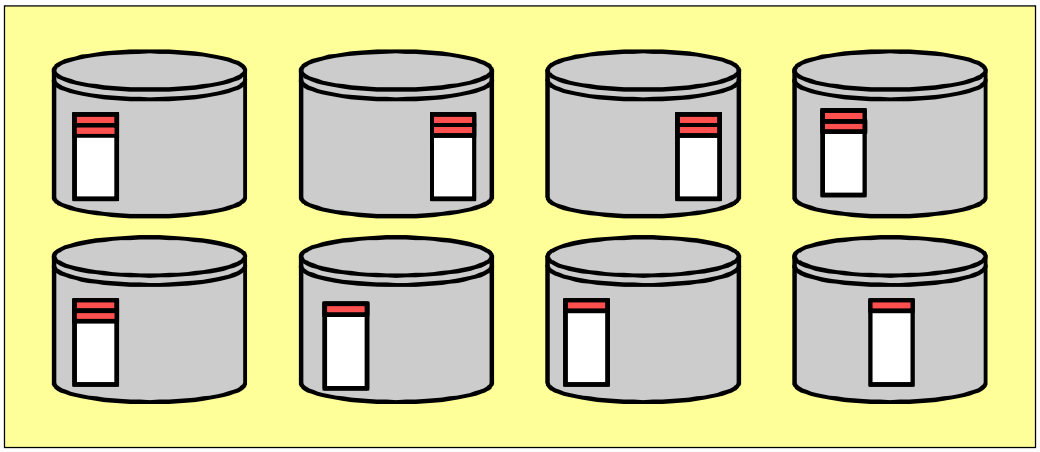
Stripe column and stripe width are properties that are set when you create the volume. The stripe column is used both for allocating the initial space for the volume but also for the way the space inside the volume is used.

The volume will be formatted for a file system. As the file system allocates space for files, the files will be stored in pieces the size of stripe width. If defaults are used, they are stored in 128 KB pieces. 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
- 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 will be created in every volume allocation unit of 512 MB. Eight volume extents are shown, one on each of the eight disks.

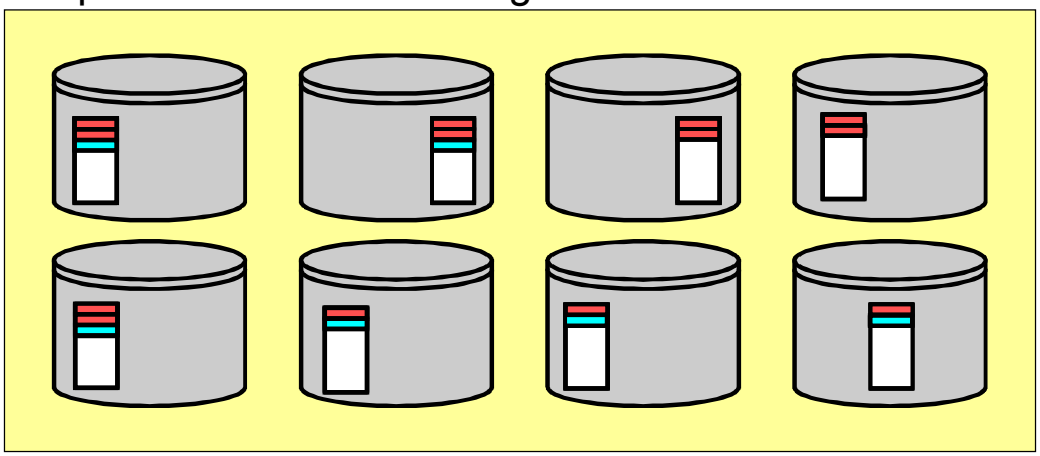
**Note:** Each VAU will be filled before the next VAU is used.

The diagram on this page shows how ACFS volume striping works. The strip width is 16 KB. A new file of 200 KB will be 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 six chunks of 16 KB stripes across the same eight allocation units



ORACLE



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 higher before you can create an ADVM volume in the disk group. You can set the attribute with the `asmcmd` command:

```
setattr -G DATA compatibile.advm 11.2.0.0.0
```

To create and mount an ACFS volume, use the procedure shown in the slide.

As a user with `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 will be 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 says add this volume 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:

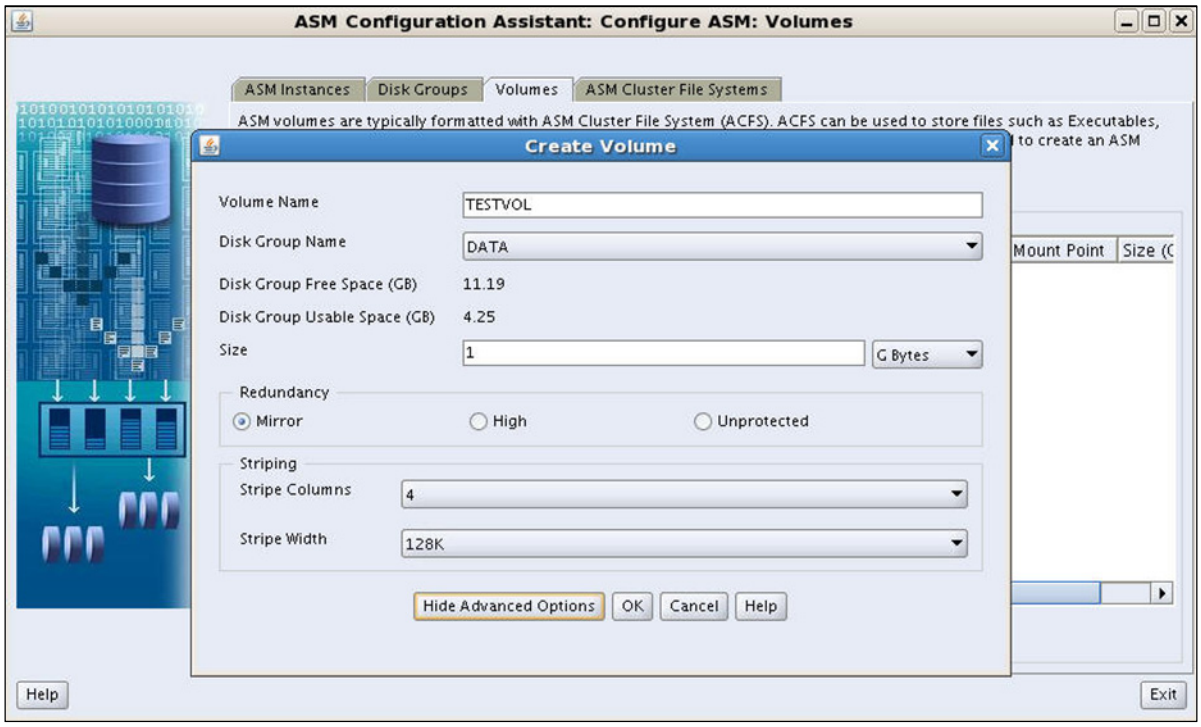
```
SQL> ALTER DISKGROUP DGROUPA
          ADD VOLUME asmvol1 SIZE 10g;
SQL> ALTER DISKGROUP DGROUPA
          RESIZE VOLUME asmvol1 SIZE 15G;
SQL> ALTER DISKGROUP DGROUPA
          DROP VOLUME asmvol1;
SQL> ALTER DISKGROUP DGROUPA
          ENABLE VOLUME asmvol1;
SQL> ALTER DISKGROUP ALL
          DISABLE 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 a part of a disk group. SQL*Plus allows you to create volumes with the `add` command, `resize` the volume, `drop`, `enable`, and `disable` the volume. The `modify volume` command allows you to set the intelligent data placement option [`HOT`|`COLD`], the mount point, and the usage name of the volume.

# Creating an ACFS Volume with ASMCA

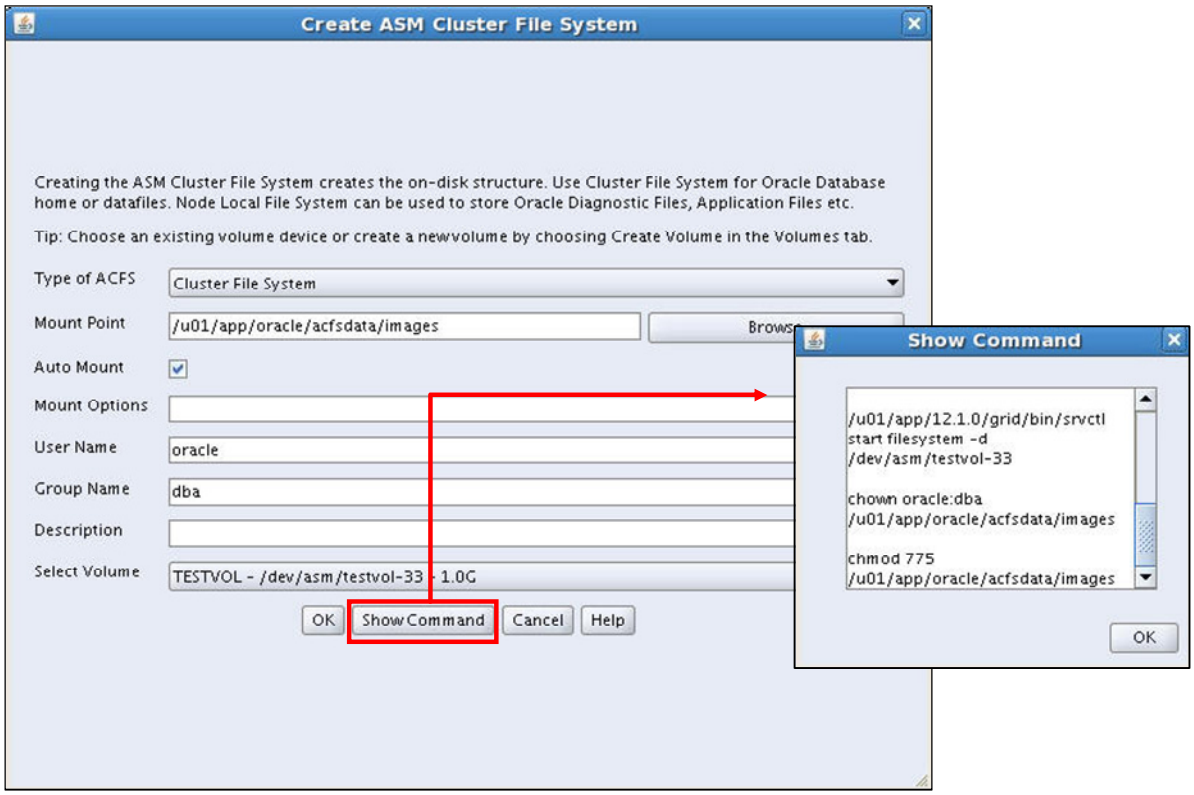





In ASMCA, click the Volume tab, and then, on the Volumes page, click Create. On the the Create Volume page, you specify the name, disk group, and size of the volume. Optionally, with the Advanced options you can specify the mirroring and striping attributes.

Click OK to create the ACFS volume.

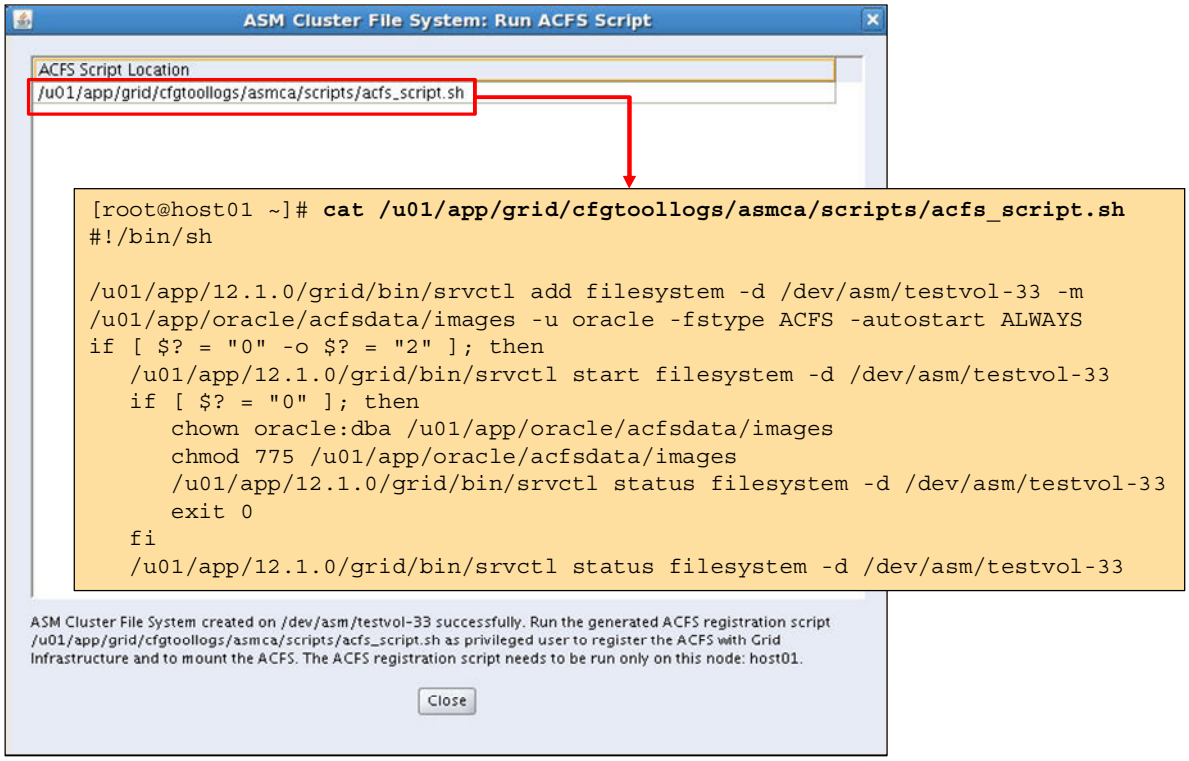
# Creating the ACFS File System with ASMCA






After the volume is created, on the Create ASM Cluster File System page, select the volume, the type of file system, and mount point. ASMCA creates the file system on the volume and registers the volume, but the mount must be done by the `root` or `administrator` user.

# Register and Mount the ACFS File System with ASMCA






On the Mount ACFS Command page, follow the instructions. Execute the command shown on the first node in the cluster as the `root` 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]
```



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 above 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 the 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 using the 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` | Display new ACFS file and file system features and information. |
| `acfs plugin disable\|enable\|info` | Manage the ACFS plug-in infrastructure |
| `acfsutil snap` | Create and delete ACFS snapshots. |
| `acfsutil registry` | Register an ACFS file system with the ACFS mount registry |
| `acfsutil rmfs` | Remove an ACFS file system. |
| `acfsutil size` | Resize an ACFS file system. |
| `acfsutil snap convert\| create\|delete\|info` | Manage snapshots |
| `acfsutil tune` | View or modify 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.

# Configuration Settings for Database Files

- Set the ASM and ADVM compatibility attributes to `12.1`
  - Required to enable new ASM and ADVM features
  - Command syntax:

```
SQL> [ CREATE | ALTER ] DISKGROUP ...
     ATTRIBUTE 'compatible.asm' = '12.1',
               'compatible.advm' = '12.1';
```

- Set stripe columns to `1` for the ADVM volume
  - Disables ADVM volume striping
  - Command syntax:

```
ASMCMD> volcreate -G diskgroup -s size --column 1 volume
```

- Set `FILESYSTEMIO_OPTIONS=SETALL` in the database initialization parameter file
  - Enables direct I/O for the database, bypassing the OS file system cache



The slide outlines the recommended configuration settings to achieve optimal performance with database data files on Cloud FS.

# Cloud FS 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|NTFS|ZFS|JFS|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
  - No difference between general file systems and file systems that contain Oracle Database home directories



Oracle Database 12*c* provides the following enhancements to cluster resources for Cloud FS:

- **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 on 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 attributes 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 previous 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 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 of 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 previous 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 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 separate node-specific file systems 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 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 the `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 associated with an application requiring 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 will also be 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 higher.

ORACLE



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 the online recovery of files inadvertently modified or deleted from a file system. With up to a total of 63 read-only, read-write, or combination of read-only and read-write snapshot views supported for each file system, flexible online file recovery solutions spanning multiple views can be employed. An Oracle ACFS snapshot can also be used as the source of a file system backup, as 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 higher. If you create a read-write snapshot on an existing Oracle ACFS file system from a version earlier than 11.2.0.3.0, then the file system is updated to the 11.2.0.3.0 or higher format. After a file system has been updated to a higher 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
```



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 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 will create 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: 12.1.0.1.0
    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
```

# Snapshot Enhancements

- Snaps-of-Snaps
  - 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 63 snapshots, including Snaps-of-Snaps.
  - 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*, Cloud FS supports the creation of snapshots based on an existing snapshot of the same ACFS file system; 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. Each ACFS file system can support a total of 63 snapshots, including Snaps-of-Snaps.

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 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 supporting native operating system file system application programming interfaces (APIs). Consequently, backup applications that access files using the native operating system file system interfaces are able to access and backup Oracle ACFS file systems and other native operating system file systems. Oracle ACFS snapshots can be dynamically created and used to present a consistent, on-line 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 upon 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.

ORACLE



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 |

ORACLE



The table above 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 sizes in Windows folders. The mechanism used by Folder Properties on Windows systems only accounts for files and should be considered an approximate value.

# Quiz

Oracle CloudFS components include:

a. ASM Cluster File System
b. Oracle Database
c. Oracle ASM Dynamic Volume Manager
d. Oracle Clusterware



**Answer: a, c**

# 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 6 Overview: Administering Oracle CloudFS

This practice covers the following topics:

- Managing an ACFS file system
  - Create
  - Register
  - Mount
- Managing ACFS Snapshots

7

# Oracle CloudFS Advanced Topics

# Objectives

After completing this lesson, you should be able to:

- Configure and manage ACFS Auditing
- Implement ACFS Encryption
- Configure and manage ACFS Replication
- Implement ACFS tagging
- Describe the ACFS Plug-in Architecture
- Configure High Availability NFS

# Cloud FS Auditing

With Oracle Database 12*c*, Cloud FS 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 Cloud FS.



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*, Cloud FS 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 Cloud FS, but provides a way for Cloud FS audit data to be integrated into Oracle Audit Vault.

Although the concept of file system auditing is not new, Cloud FS auditing delivers the following set of advanced capabilities:

- Cloud FS auditing provides the same functionality and management interfaces across all of the platforms that Cloud FS supports.
- Cloud FS auditing enforces separation of duties for audit trail management and review.
- Cloud FS auditing integrates with Oracle Audit Vault, which provides a secure offline store for audit information, along with analysis and reporting capabilities.

# Cloud FS 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



The Cloud FS audit trail is contained in a set of files inside Cloud FS. Audit trail files generated by Cloud FS auditing are designed for:

- Manual review by a Cloud FS auditor 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 Cloud FS 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                                          Header
Encoding: UTF-8

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                                                File Access Event
Process: 4567
Host: host1
File: /my_mount_point1/hr_data/payroll
Evaluation Result: ACFS_REALM_AUTH
Realm: myPayrollRealm
Application: vi

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                                              Privilege Use Event
Group: 1823
Process: 8901
Host: host1
Command Line: acfsutil sec prepare -m /my_mount_point2
```

ORACLE



The Cloud FS audit trail consists of a set of audit records. Each audit record represents a single event.

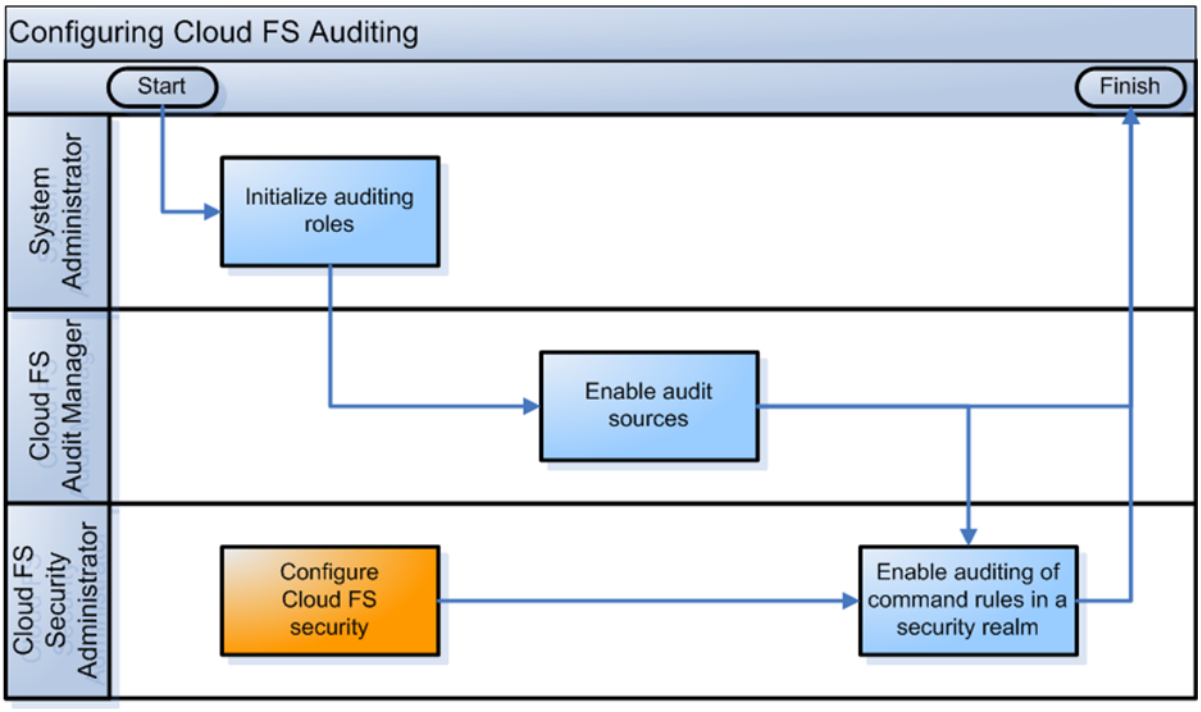
The audit trail has a brief header at the beginning of each file. The header identifies the version of the audit file format and the character encoding for the audit file.

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 Cloud FS Auditing






The diagram on the slide illustrates the process for configuring Cloud FS auditing. It outlines the key process tasks and who is required to perform them.

Note that the task of configuring Cloud FS security is separate from configuring Cloud FS 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 Cloud FS and creating ACFS file systems.

Further detail regarding the steps for configuring Cloud FS auditing are provided on the following pages.

# Initializing Auditing Roles and Enabling Audit Sources

- Set up 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 a Cloud FS 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 Cloud FS audit manager or Cloud FS 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 Cloud FS security (`-s sec`) or Cloud FS encryption (`-s encr`). Auditing can be enabled for both Cloud FS security and Cloud FS 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 a Cloud FS security realm

- Optional task
- Performed by a Cloud FS security administrator
- Command syntax:

```
$ acfsutil sec realm audit enable Realm -m Mount_Point
  [-l Command_Rule,Command_Rule,...] [-a] [-v [-u]]
```



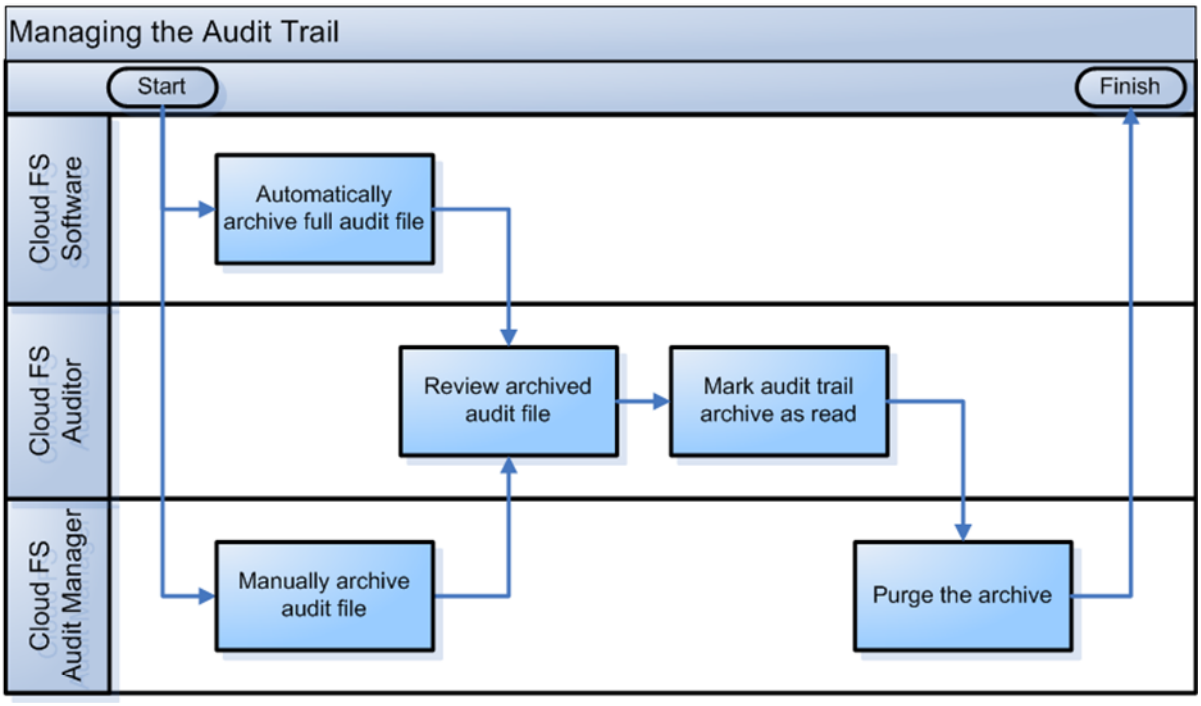
In addition to the core auditing that is enabled by the `acfsutil audit enable` command, auditing of specific command rules in a Cloud FS security realm can be enabled by using the `acfsutil sec realm audit enable` command. Note that only the security administrator, not the audit manager, can run this 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 regarding Cloud FS security, including realms and command rules, refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)*.

Note that Cloud FS 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 on the slide illustrates the process for managing the Cloud FS 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 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 a Cloud FS audit manager.
  - Enables immediate review of recent audit data
  - 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, Cloud FS 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 has been 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 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]
```



Purging removes 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 data in an ACFS file system to prevent unauthorized data use in 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 type of encryption keys:
  - File Encryption Key: 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.
- Backup the OCR after creating an encryption key to ensure there is a backup containing all of the volume encryption keys for the file system.



Oracle ACFS encryption enables you to encrypt data stored on disk (data-at-rest). The encryption feature protects data in an Oracle ACFS file system in encrypted format 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 requires system administrator privileges. This functionality includes the 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 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, then specify 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 the file data.

You should back up the Oracle Cluster Registry (OCR) after creating or updating an encryption key to ensure there is an OCR backup that contains all of 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, files 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 higher For Linux
  - 11.2.0.3 or higher For Windows
- Encryption information for ACFS file systems is displayed in the V$ASM_ACFS_ENCRYPTION_INFO view.



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 has been added to a security realm, all files 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 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 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 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 `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 pre-defined 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 higher. The disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or higher 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 higher.

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:

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:

1. Initialize encryption. Run the `acfsutil encr init` command to initialize encryption and create the storage necessary for the encryption keys. This command must be run one time 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 mounted on `/acfsmnts/acfs1`:

```
# /sbin/acfsutil encr set -a AES -k 128 -m /acfsmnts/acfs1/
```

The command above also transparently generates a volume encryption key which is stored in the key store that was previously configured with the `acfsutil encr init` command.

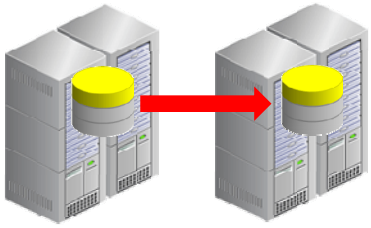
3. Enable encryption. Run `acfsutil encr on` to enable encryption for directories and files. For example, the following command enables encryption recursively on all files in the `/acfsmnts/acfs1/medrecords` directory.

```
# /sbin/acfsutil encr on -r /acfsmnts/acfs1/medrecords -m /acfsmnts/acfs1/
```

For users that have permission to access files in the `medrecords` directory, they 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, 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 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



Oracle ACFS replication enables replication of Oracle ACFS file systems across the network to a remote site, providing disaster-recovery capability for the file system. Oracle ACFS replication can only be configured 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 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 ACFS file system cannot be used as a primary and a standby file system.

Oracle ACFS replication captures file system changes written to disk for a primary file system and records the changes in files called replication logs. These logs are transported to the site hosting 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 hosting the primary and standby file systems.

It is critical that there is enough disk space 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 only remove files that are not being replicated because the 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 has 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 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 hosting 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 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 necessary for the replication logs 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 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 higher 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 higher. 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 higher.

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:

1. Determine the storage capacity necessary for replication on the sites 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 being replicated and the space necessary for the replication logs sent from the primary file system. Calculate the replication-related storage requirement for the primary file system, 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 hosting 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 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 hosting the standby file system by performing the following procedures.

Create a new file system of adequate size to hold the replicated files and 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 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, then use 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 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 replication background processes.



4. Configure the site 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 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, then use the `-c` option.
- If the mount point is different on the site hosting the standby file system than it is on the site hosting the primary file system, specify the mount point with the `–m` option.

5. Monitor information about replication on the file system. The `acfsutil repl info` command displays information about the state of the replication processing on either system.

6. Manage replication background processes. Run the `acfsutil repl bg` command to start, stop, or retrieve information about replication background processes.

Run the `acfsutil repl pause` to momentarily stop replication, if needed. You should run `acfsutil repl resume` command soon as possible to resume replication.

# Using Replication in Conjunction with Cloud FS Security and Encryption

With Oracle Database 12*c*, Cloud FS replication can be used in conjunction with Cloud FS 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*, Cloud FS replication can be used in conjunction with Cloud FS 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 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 is not altered.

# ACFS Tagging

- ACFS tagging assigns a common naming attribute to a group of files.
- ACFS Replication can use tags to select files with a unique tag name for replication to a different remote cluster site.
- The tagging option avoids having to replicate an entire ACFS file system.
- ACFS implements tagging with Extended Attributes.
- To use ACFS tagging, set the compatibility attributes for ASM and ADVM to:
  - 11.2.0.2 or higher for Linux
  - 11.2.0.3 or higher for Windows
  - 12.1 or higher for Solaris and AIX

ORACLE



ACFS tagging assigns a common naming attribute to a group of files. ACFS Replication can use this tag to select files with a unique tag name for replication to a different remote cluster site. The tagging option avoids having to replicate an entire ACFS file system.

ACFS implements tagging with Extended Attributes. Some editing tools and backup utilities do not retain the Extended Attributes of the original file by default; you must set a specific switch.

To use Oracle ACFS tagging functionality on Linux, the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.2 or higher. To use Oracle ACFS tagging functionality on Windows, the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or higher. To use Oracle ACFS tagging functionality on Solaris or AIX, the disk group compatibility attributes for ASM and ADVM must be set to 12.1 or higher.

# Tagging ACFS File Systems

Managing ACFS tags:

1. Specify tag names for directories and files.

```
$ acfsutil tag set -r comedy /acfsmnts/repl_data/films/comedies
$ acfsutil tag set -r drama /acfsmnts/repl_data/films/dramas
$ acfsutil tag set -r drama /acfsmnts/repl_data/films/mysteries
```

2. Display tagging information.

```
$ acfsutil tag info -r /acfsmnts/repl_data/films
$ acfsutil tag info -t drama -r /acfsmnts/repl_data/films
```

3. Remove and change tag names if necessary.

```
$ acfsutil tag unset -r drama /acfsmnts/repl_data/films/mysteries
$ acfsutil tag set -r mystery /acfsmnts/repl_data/films/mysteries
```

ORACLE



The steps to manage tagging are:

1. Specify tag names for directories and files. Run the `acfsutil tag set` command to set tags on directories or files. You can use these tags to specify which objects are replicated. For example, add the comedy and drama tags to the files in the subdirectories of the `/acfsmnts/repl_data/films` directory.

```
$ acfsutil tag set -r comedy /acfsmnts/repl_data/films/comedies
$ acfsutil tag set -r drama /acfsmnts/repl_data/films/dramas
$ acfsutil tag set -r drama /acfsmnts/repl_data/films/mysteries
```

In this example, the `drama` tag is purposely used twice and that tag is changed in a later step.

2. Display tagging information. Run the `acfsutil tag info` command to display tag names for directories or files in ACFS file systems. Files without tags are not displayed. For example, display tagging information for files in the `/acfsmnts/repl_data/films` directory.

```
$ acfsutil tag info -r /acfsmnts/repl_data/films
```

Display tagging information for files with the `drama` tag in the `/acfsmnts/repl_data/films` directory.

```
$ acfsutil tag info -t drama -r /acfsmnts/repl_data/films
```

3. Remove and change tag names if necessary. Run the `acfsutil tag unset` command to remove tags on directories or files. For example, unset the `drama` tag on the files in the `mysteries` subdirectory of the `/acfsmnts/repl_data/films` directory to apply a different tag to the subdirectory.

```
$ acfsutil tag unset -r drama /acfsmnts/repl_data/films/mysteries
```

Add the `mystery` tag to the files in the `mysteries` subdirectory of the `/acfsmnts/repl_data/films` directory.

```
$ acfsutil tag set -r mystery /acfsmnts/repl_data/films/mysteries
```

# Generic API for Cloud FS Tagging

Cloud FS provides an API for Cloud FS tagging:

- API compliments existing command line interfaces.
- API is generic and platform independent.
  - Implemented as a C library
- API provides the following operations:
  - `acfsgettag` Determines if a file contains a tag
  - `acfslisttags` Lists tags that are assigned to a file
  - `acfsremovetag` Removes a tag from a file
  - `acfssettag` Adds a tag to a file



Oracle Database 12*c* provides a generic, platform-independent API to support tagging operations on files. The API compliments the existing `acfsutil tag` commands that are available in previous releases. The API is implemented as a C library and made available as part of the standard Oracle Database 12*c* software distribution.

The slide outlines the operations that are provided by the API. Essential documentation for the API can be found in the header file at `$ORACLE_HOME/usm/public/acfslib.h`. A demonstration application using the tagging API is also available at `$ORACLE_HOME/usm/demo`. Refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)* for further information.

# Cloud FS Plug-in Infrastructure

The Cloud FS plug-in infrastructure enables user applications to access file system and volume metrics:

- Allows for customized monitoring solutions that include ACFS file system and volume data
- Can be enabled on individual file systems and hosts
- Referenced by applications using the Cloud FS plug-in API
  - API supports message posting and polling delivery models



With Oracle Database 12*c*, Cloud FS provides infrastructure that enables a user space application to access just-in-time ACFS file system and ADVM volume metrics. Applications can use the Cloud FS plug-in infrastructure to create customized solutions that extend the general application file metric interfaces to include detailed file system and volume data.

The Cloud FS plug-in can be enabled on individual ACFS file systems mounted on a standalone host, or on one or more nodes of a cluster where the Oracle ACFS file system is mounted. This enables message communication between an ACFS file system and an associated user space application module by using the Cloud FS plug-in application programming interface (API).

The plug-in API supports both polling and posting message delivery models and multiple message payload types.

Essential documentation for the API can be found in the header file at `$ORACLE_HOME/usm/public/acfslib.h`. Refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)* for detailed information regarding the plug-in infrastructure and API.

# Using the Cloud FS Plug-in

- Enabling the plug-in:

```
$ acfsutil plugin enable [tag ...] metrictype [interval[s|m]]
  mount_point
```

```
$ acfsutil plugin enable HRDATA acfsmetric1 /hr
```

```
$ acfsutil plugin enable acfsmetric1 5m /sales
```

- Referencing the metrics:

```
#include <usacfslib.h>

/* allocate message buffers */
ACFS_METRIC1 *metrics = malloc (sizeof(ACFS_METRIC1));

/* poll for metric1 data */
rc = acfsplugin_metrics(ACFS_METRIC_TYPE1, metrics,
sizeof(metrics), mountp);

/* print message data */
printf("reads %8llu ", metrics->acfs_nreads);
printf("writes %8llu ", metrics->acfs_nwrites);
printf("avg read size %8u ", metrics->acfs_avgrsize);
```

ORACLE



To use the Cloud FS plug-in API, the plug-in must first be enabled. The slide shows the general syntax for the `acfsutil plugin enable` command, followed by two examples.

In the first example, the plug-in is enabled for all files in the `/hr` file system that are tagged with the `HRDATA` tag. In this case, no interval is specified, which implies that the metrics will be referenced by using the polling delivery model.

In the second example, the plug-in is enabled for all files in the `/sales` file system, regardless of tagging. In this case, a five-minute interval is also specified, which implies that the metrics will be referenced by using message posting as the delivery model.

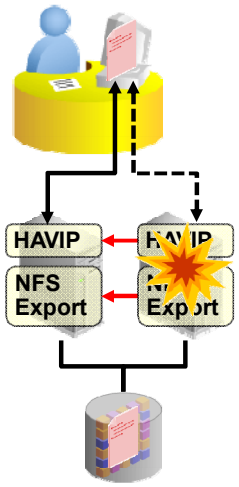
In both examples, `acfsmetric1` is specified as the metric type. This predefined metric type contains a collection of metrics, including number of reads, number of writes, average read size, average write size, minimum and maximum read size, minimum and maximum write size, read throughput (bytes per second) and write throughput.

After the plug-in is enabled, user applications can reference the exposed metrics. The example at the bottom of the slide shows some code fragments. The `acfsplugin_metrics` function reads the metrics associated with the file system specified in the `mountp` argument. If the plug-in was enabled without an interval, the function call polls for the latest metrics. If the plug-in was enabled with an interval, the call to `acfsplugin_metrics` is blocked, and the application pauses until the metrics are next posted.

# High Availability NFS

High Availability NFS (HANFS) provides an 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.






With Oracle Database 12*c*, Cloud FS 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
```



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 on 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), not currently in use, and 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.

# Miscellaneous Cloud FS Enhancements

- Support for cluster-wide, file-granular advisory file locking by using the POSIX `fcntl` interface
  - Byte range locks continue to operate in node local mode.
- End-to-end storage visibility for files

```
$ acfsutil info file -d filename
```

  - Displays detailed file information, including file extent location on the Oracle ASM devices
- Improved directory listing performance for newly created directories
- Unicode support for file names on Windows
  - Unicode file names are already supported on other platforms.



The slide lists other miscellaneous Cloud FS new features in Oracle Database 12*c*.

- Oracle ACFS supports clusterwide, file granular `fcntl` advisory file locking while byte range locks continue to operate in node local mode.
- Oracle ACFS directory listing performance is improved for newly created directories.
- Oracle ACFS supports end-to-end storage visibility for files with the `-d` option of `acfsutil info file`. The `-d` option displays details about a file extent location, from the Oracle ACFS extent to the Oracle ASM devices in the disk group.
- Oracle ACFS supports unlimited expansions when resizing a file system in a disk group with ADVM compatibility set to 11.2.0.4 or higher.

# Quiz

Identify the correct statements regarding Cloud FS support for storing Oracle Database files:

a. With Oracle Database 12*c*, Cloud FS is supported as a store for all Oracle Database files.
b. A Cloud FS snapshot can be used as a point-in-time image for backups while database processing continues uninterrupted.
c. Cloud FS replication can be used to replicate database files for disaster recovery purposes.

ORACLE



**Answer: a, b**

With Oracle Database 12*c*, support will not be available for Oracle Database files on file systems that use Cloud FS replication. Oracle Data Guard remains the recommended Oracle Database replication technology for disaster protection.

# Quiz

Identify the correct statements regarding Cloud FS file system resources:

a. A file system resource can specify a list of nodes on which the file system will be mounted.
b. A file system resource can specify a list of server pools on which the file system will be mounted.
c. All file system attributes are recorded in the file system resource, removing the need for the ACFS mount registry.
d. 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.

ORACLE



**Answer: a, b, c, d**

Cloud FS file system resources display all of the attributes listed in the question.

# Summary

In this lesson, you should have learned how to:

- Configure and manage ACFS Auditing
- Implement ACFS Encryption
- Configure and manage ACFS Replication
- Implement ACFS tagging
- Describe the ACFS Plug-in Architecture
- Configure High Availability NFS

# Practice 7 Overview: Oracle CloudFS Advanced Topics

This practice covers the following topics:

- Configuring and Using HANFS.
- Configuring and Using Cloud FS Auditing.
- Implementing Node-Specific File System Dependencies.