Oracle® Database

Storage Administrator's Guide

11*g* Release 1 (11.1)

B31107-06

February 2012

Oracle Database Storage Administrator's Guide, 11*g* Release 1 (11.1)

B31107-06

Contributing Authors: Prasad Bagal, Mark Bauer, Eric Belden, Steve Fogel, Rich Long, Colin McGregor, Hanlin Qian, Kathy Rich, Ara Shakian, Amar Subba, Michael Timpanaro-Perrotta, Radek Vingralek, Nitin Vengurlekar, Ron Weiss, Rajiv Wickremesinghe, Jim A. Williams, Song Ye, Krishnan Yegnashankaran, Hector Yuen

Contents

3 Administering ASM Instances

4 Administering ASM Disk Groups

6 Administering ASM with Oracle Enterprise Manager

7 ASM Command-Line Utility

A Deploying ASM Using SQL*Plus

Glossary

Index

Preface

The *Oracle Database Storage Administrator's Guide* describes how to administer Oracle Automatic Storage Management (Oracle ASM) for Oracle databases. This Preface contains the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

The audience for this book includes system administrators, database administrators, and storage administrators. The *Oracle Database Storage Administrator's Guide* is intended for database and storage administrators who perform the following tasks:

- Administer and manage Oracle ASM
- Configure and administer Oracle ASM

To use this document, you should be familiar with basic Oracle Database concepts and administrative procedures. Also, you might want to review the documentation for Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC). See "Related Documents" on page x.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Related Documents

For more information, refer to the following Oracle resources:

- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day + Real Application Clusters Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database Concepts*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- Platform-specific guides, including Oracle Database, Oracle Grid Infrastructure, and Oracle Real Application Clusters installation guides

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in Oracle Database Storage Administration?

This section describes the new storage administration features for Oracle Database 11*g* release 1 (11.1) and it contains the following topic:

- New Automatic Storage Management Features for Oracle Database 11g Release 1 (11.1)

> **See Also:** *Oracle Database New Features Guide* for a complete description of the new features in Oracle Database 11*g* release 1 (11.1)

New Automatic Storage Management Features for Oracle Database 11*g* Release 1 (11.1)

This section describes the following Oracle Database 11*g* release 1 (11.1) Automatic Storage Management (ASM) features:

- *Oracle Database Storage Administrator's Guide* is New for this Release

  This book, the *Oracle Database Storage Administrator's Guide*, is new for Oracle Database 11*g* release 1 (11.1) and it is the primary information source for Oracle Automatic Storage Management features.

- ASM Fast Mirror Resync

  ASM fast mirror resync quickly resynchronizes ASM disks within a disk group after transient disk path failures as long as the disk drive media is not corrupted. Any failures that render a failure group temporarily unavailable are considered transient failures. Disk path malfunctions, such as cable disconnections, host bus adapter or controller failures, or disk power supply interruptions, can cause transient failures.

  The duration of a fast mirror resync depends on the duration of the outage. The duration of a resynchronization is typically much shorter than the amount of time required to completely rebuild an entire ASM disk group.

  > **See Also:** "ASM Fast Mirror Resync" on page 4-17 for more information about ASM fast mirror resync

- ASM Rolling Upgrade

  You can now place an ASM Cluster in rolling upgrade mode, which enables you to operate with mixed ASM versions starting with Oracle Database 11*g* release 1

(11.1) and later. As a result, ASM nodes can be independently upgraded or patched without affecting database availability.

> **See Also:** "Using ASM Rolling Upgrades" on page 3-15 for more information about ASM rolling upgrade

- New SYSASM Privilege and OSASM operating system group for ASM Administration

  This feature introduces a new SYSASM privilege that is specifically intended for performing ASM administration tasks. Using the SYSASM privilege instead of the SYSDBA privilege provides a clearer division of responsibility between ASM administration and database administration.

  OSASM is a new operating system group that is used exclusively for ASM. Members of the OSASM group can connect as `SYSASM` using operating system authentication and have full access to ASM.

  > **See Also:** "Authentication for Accessing ASM Instances" on page 3-17 for more information about the SYSASM privilege

- ASM Scalability and Performance Enhancements

  ASM file extent management has been enhanced to improve performance and to use significantly less SGA memory to store file extents. When ASM files increase in size, the size of each new extent also increases automatically so that fewer extent pointers are required to describe the file. This feature improves performance when accessing ASM files that are 20 GB and larger, up to 128 TB. Very large databases (VLDBs) often require these large file sizes.

  In addition, when you create new disk groups you now have multiple allocation unit size options such as 1, 2, 4, 8, 16, 32, and 64. You might obtain significant performance improvements by selecting larger AUs depending on the type of workloads (typically large sequential I/O) and storage system types.

  > **See Also:** "Extents" on page 1-7 for more information about scalability and performance enhancements

- New ASM Command Line Utility (ASMCMD) Commands and Options

  ASMCMD has the following four new commands: `lsdsk`, `md_backup`, `md_restore` and `remap`. In addition, you can use new options for the `ls` and `lsdg` commands. The following describes the four new ASM commands:

  – `cp`—Enables you to copy files between ASM disk groups on local instances and remote instances.

  – `lsdsk`—ASM can list disk information with or without a running ASM instance. This is a useful tool for system or storage administrators who want to obtain lists of disks that an ASM instance uses.

  – `md_backup` and `md_restore`—These commands enable you to re-create a pre-existing ASM disk group with the same disk path, disk name, failure groups, attributes, templates and alias directory structure. You can use `md_backup` to back up the disk group environment and use md_restore to re-create the disk group before loading from a database backup.

  – `remap`—You can remap and recover bad blocks on an ASM disk in normal or high redundancy that have been reported by storage management tools such as disk scrubbers. ASM reads from the good copy of an ASM mirror and rewrites these blocks to an alternate location on disk.

> **See Also:** "ASMCMD Command Reference" on page 7-7 for more information about new and enhanced ASMCMD commands

- Enhancements to `ALTER DISKGROUP`, `CREATE DISKGROUP`, and `DROP DISKGROUP` to manage Automatic Storage Management disk groups

  The `ALTER DISKGROUP`, `CREATE DISKGROUP`, and `DROP DISKGROUP` SQL statements have been enhanced with additional options.

  – `CREATE DISKGROUP` and `ALTER DISKGROUP` have new syntax that lets you set various attributes of a disk group.

  – The `CHECK` clause of `ALTER DISKGROUP` has simplified syntax for checking the consistency of disk groups, disks, and files in an Automatic Storage Management environment.

  – The `MOUNT` clause of `ALTER DISKGROUP` offers new options when mounting a disk group.

  – The `ONLINE` and `OFFLINE` clauses of `ALTER DISKGOUP` let you take a disk offline for repair and then bring it back online.

  – The `FORCE` keyword of `DROP DISKGROUP` lets you drop a disk group that can no longer be mounted by an ASM instance.

  > **See Also:** Chapter 4, "Administering ASM Disk Groups" for more information about administering disk groups with the `ALTER DISKGROUP`, `CREATE DISKGROUP`, and `DROP DISKGROUP` SQL statements.

- New Attributes for Disk Group Compatibility

  To enable some of the new ASM features, you can use two new disk group compatibility attributes, `COMPATIBLE.RDBMS` and `COMPATIBLE.ASM`. These attributes specify the minimum software version that is required to use disk groups for the database and for ASM respectively. This feature enables heterogeneous environments with disk groups from both Oracle Database 10*g* and Oracle Database 11*g*. By default, both attributes are set to `10.1`. You must advance these attributes to take advantage of the new features.

  > **See Also:** "Disk Group Compatibility" on page 4-21 for more information about disk group compatibility

- ASM Preferred Read Failure Groups

  This feature is useful in extended clusters where remote nodes have asymmetric access with respect to performance. This enables more efficient use of network resources by eliminating the need to use the network to perform read operations.

  ASM in Oracle Database 10*g* always reads the primary copy of a mirrored extent set. In Oracle Database 11*g*, when you configure ASM failure groups it might be more efficient for a node to read from a failure group that is closest to the node, even if it is a secondary extent. You can configure your database to read from a particular failure group extent by configuring preferred read failure groups.

  > **See Also:** "Preferred Read Failure Groups" on page 4-19 for more information about preferred read failure groups

- ASM Fast Rebalance

Rebalance operations that occur while a disk group is in `RESTRICTED` mode eliminate the lock and unlock extent map messaging between ASM instances in Oracle RAC environments, improving overall rebalance throughput.

> **See Also:** "About Restricted Mode" on page 3-12 for more information about ASM fast rebalance

1

Introduction to Automatic Storage Management (ASM)

This chapter describes Automatic Storage Management (ASM) concepts and provides an overview of ASM features. This chapter contains the following topics:

- Overview of Oracle Automatic Storage Management (ASM)
- Understanding ASM Concepts
- ASM Disk Group Administration

For a list of the terms that are used in the *Oracle Database Storage Administrator's Guide* and their definitions, refer to the Glossary in this guide.

Overview of Oracle Automatic Storage Management (ASM)

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. ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices.

ASM uses **disk group**s to store datafiles; an ASM disk group is a collection of disks that ASM manages as a unit. Within a disk group, ASM exposes a file system interface for Oracle database files. The content of files that are stored in a disk group are evenly distributed, or striped, to eliminate hot spots and to provide uniform performance across the disks. The performance is comparable to the performance of raw devices.

You can add or remove disks from a disk group while a database continues to access files from the disk group. When you add or remove disks from a disk group, ASM automatically redistributes the file contents and eliminates the need for downtime when redistributing the content.

The ASM volume manager functionality provides flexible server-based mirroring options. The ASM normal and high redundancy disk groups enable two-way and three-way mirroring respectively. You can use external redundancy to enable a Redundant Array of Inexpensive Disks (RAID) storage subsystem to perform the mirroring protection function.

ASM also uses the Oracle Managed Files (OMF) feature to simplify database file management. OMF automatically creates files in designated locations. OMF also names files and removes them while relinquishing space when tablespaces or files are deleted.

ASM reduces the administrative overhead for managing database storage by consolidating data storage into a small number of disk groups. This enables you to consolidate the storage for multiple databases and to provide for improved I/O performance.

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

Oracle Enterprise Manager includes a wizard that enables you to migrate non-ASM database files to ASM. ASM also has easy to use management interfaces such as SQL*Plus, the ASMCMD command-line interface, and Oracle Enterprise Manager.

See Also:

- Chapter 6, "Administering ASM with Oracle Enterprise Manager" for information about using Oracle Enterprise Manager
- Chapter 7, "ASM Command-Line Utility" for information about the ASMCMD command-line interface
- *Oracle Database Administrator's Guide* for information about the Oracle database structure and storage

Understanding ASM Concepts

This section describes concepts for the key ASM components and it contains the following topics:

- About ASM Instances
- About ASM Disk Groups
- About Mirroring and Failure Groups
- About ASM Disks
- About ASM Files

See Also: Chapter 2, "Preparing Storage for ASM" for more details about preparing your storage environment

About ASM Instances

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

ASM metadata is the information that ASM uses to control a disk group and the metadata resides within the disk group. ASM metadata includes the following information:

- The disks that belong to a disk group
- The amount of space that is available in a disk group
- The filenames of the files in a disk group
- The location of disk group datafile **data extent**s
- A redo log that records information about atomically changing data blocks

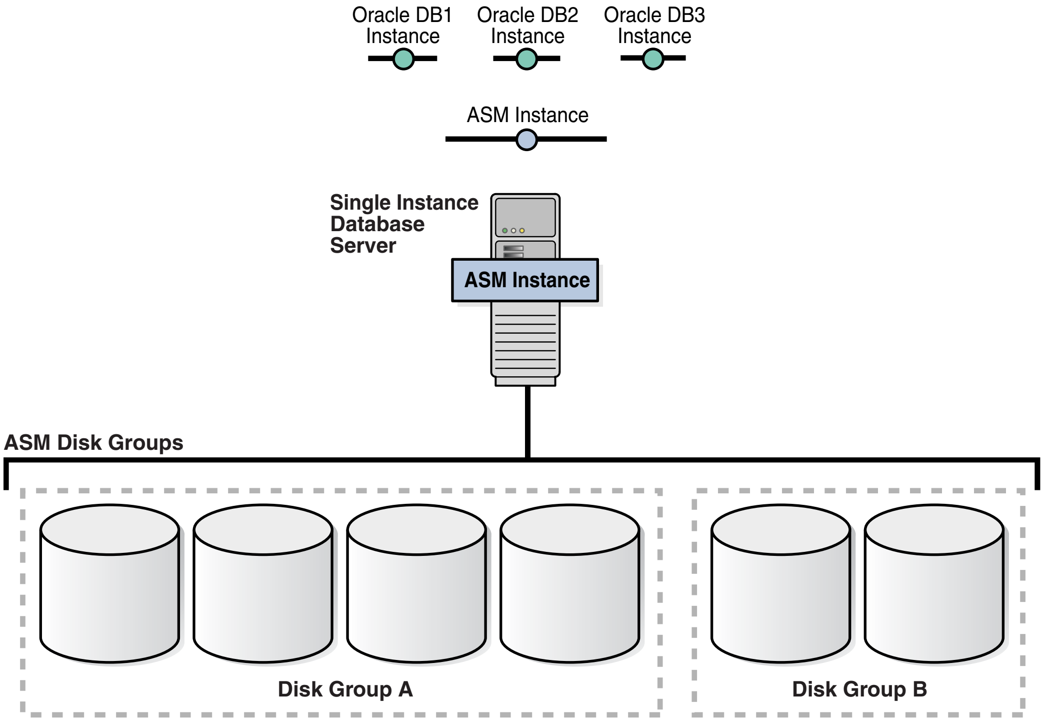
ASM and database instances require shared access to the disks in a disk group. ASM instances manage the metadata of the disk group and provide file layout information to the database instances.

ASM instances can be clustered using Oracle Clusterware; there is one ASM instance for each cluster node. If there are several database instances for different databases on the same node, then the database instances share the same single ASM instance on that node.

If the ASM instance on a node fails, then all of the database instances on that node also fail. Unlike a file system failure, an ASM instance failure does not require restarting the operating system. In an Oracle RAC environment, the ASM and database instances on the surviving nodes automatically recover from an ASM instance failure on a node.

Figure 1–1 shows a single node configuration with one ASM instance and multiple database instances. The ASM instance manages the metadata and provides space allocation for the ASM files. 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.

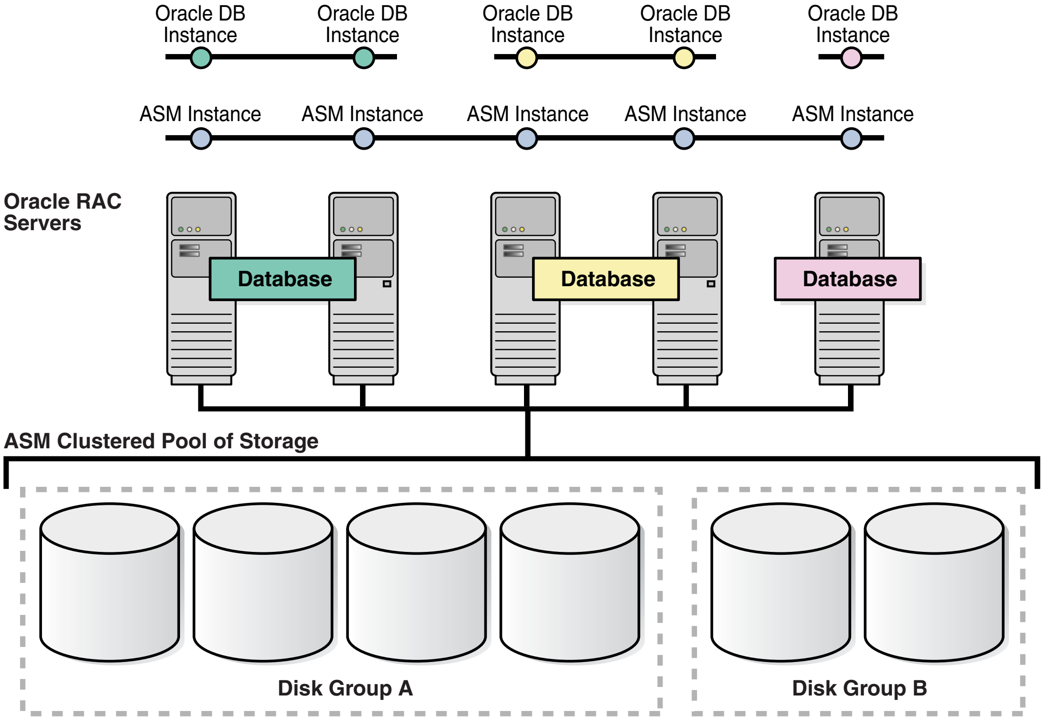
Figure 1–1 ASM for Single-Instance Oracle Databases



In Figure 1–1, there are two disk groups: one disk group has four disks and the other has two disks. The database can access both disk groups. The configuration in Figure 1–1 shows multiple database instances, but only one ASM instance is needed to serve the multiple database instances.

Figure 1–2 shows an ASM cluster in an Oracle RAC environment where ASM provides a clustered pool of storage. There is one ASM instance for each node serving multiple Oracle RAC or single-instance databases in the cluster. All of the databases are consolidated and sharing the same two ASM disk groups.

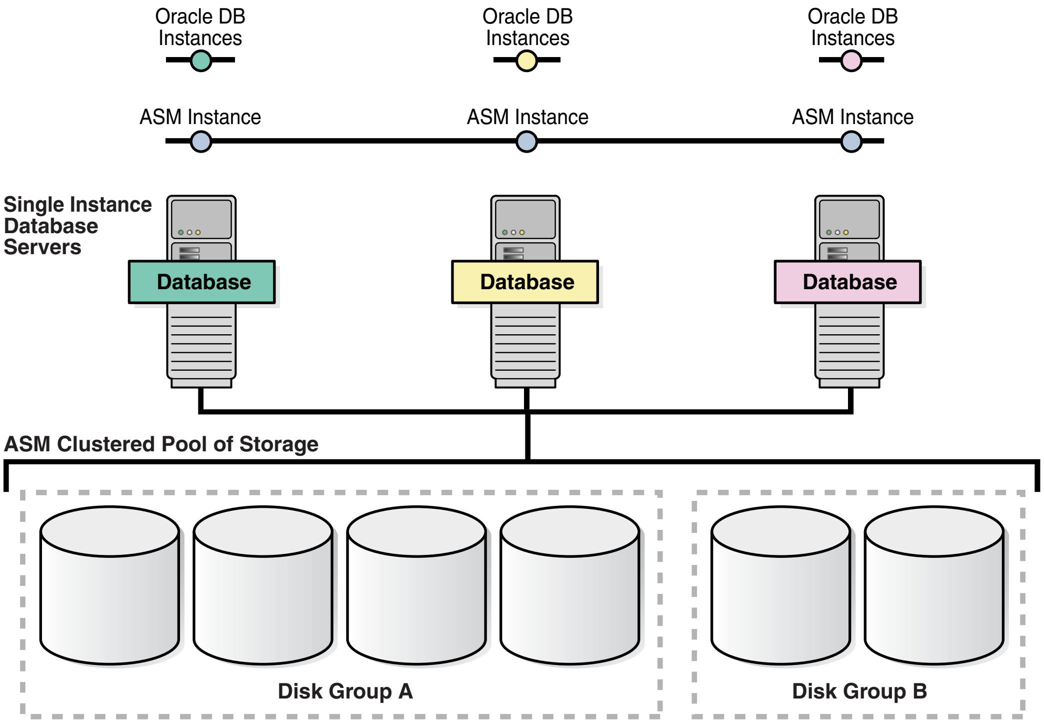
Figure 1–2 ASM Cluster Configuration with Oracle RAC



A clustered storage pool can be shared by multiple single-instance Oracle Databases as shown in Figure 1–3. In this case, multiple databases share common disk groups. A shared ASM storage pool is achieved by using Oracle Clusterware. However, in such environments an Oracle RAC license is not required.

ASM instances that are on separate nodes do not need to be part of an ASM cluster and do not communicate with each other. However, multiple nodes that are not part of an ASM cluster cannot share a disk group. To share a disk group among multiple nodes, you must install Oracle Clusterware on all of the nodes, regardless of whether you install Oracle RAC on the nodes.

Figure 1–3 ASM Cluster with Single-Instance Oracle Databases



About ASM Disk Groups

A disk group consists of multiple disks and is the fundamental object that ASM manages. Each disk group contains the metadata that is required for the management of space in the disk group.

Files are allocated from disk groups. Any 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.

Disk group components include disks, files, and allocation units. Figure 1–4 on page 1-8 shows the relationships among ASM disk group components.

About Mirroring and Failure Groups

Mirroring protects data integrity by storing copies of data on multiple disks. The disk group type determines the mirroring levels with which Oracle creates files in a disk group.

When you create a disk group, you specify an ASM disk group type based on one of the following three redundancy levels:

- **Normal** for 2-way mirroring
- **High** for 3-way mirroring
- **External** to not use ASM mirroring, such as when you configure hardware RAID for redundancy

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

ASM mirroring is more flexible than traditional RAID mirroring because you can specify the redundancy level for each file. 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, ASM allocates a primary copy and a secondary copy. ASM chooses the disk on which to store the secondary copy in a different failure group other than the primary copy. Failure groups are used to place mirrored copies of data so that each copy is on a disk in a different failure group. The simultaneous failure of all disks in a failure group does not result in data loss.

You define the failure groups for a disk group when you create an ASM disk group. After a disk group is created, you cannot alter the redundancy level of the disk group. To change the redundancy level of a disk group, create another disk group with the appropriate redundancy and then move the files to the new disk group. Oracle recommends that you create failure groups of equal size to avoid space imbalance and uneven distribution of mirror data.

If you omit the failure group specification, then ASM automatically places each disk into its own failure group. Normal redundancy disk groups require at least two failure groups. High redundancy disk groups require at least three failure groups. Disk groups with external redundancy do not use failure groups.

> **See Also:** "ASM Mirroring and Failure Groups" on page 4-13 for more information about mirroring and failure groups

About ASM Disks

ASM disks are the storage devices that are provisioned to ASM disk groups. Examples of ASM disks include:

- A disk or partition from a storage array
- An entire disk or the partitions of a disk
- Logical volumes
- Network-attached files (NFS)

When you add a disk to a disk group, you either assign a disk name or the disk is given an ASM disk name automatically. This name is different from the name used by the operating system. In a cluster, a disk may be assigned different operating system device names on different nodes, but the disk has the same ASM disk name on all of the nodes. In a cluster, an ASM disk must be accessible from all of the instances that share the disk group.

If the disks are the same size, then ASM spreads the files evenly across all of the disks in the disk group. This allocation pattern maintains every disk at the same capacity level and ensures that all of the disks in a disk group have the same I/O load. Because ASM load balances among all of the disks in a disk group, different ASM disks should not share the same physical drive.

Allocation Units

Every ASM disk is divided into allocation units (AU). An AU is the fundamental unit of allocation within a disk group. A file extent consists of one or more AU. An ASM file consists of one or more file extents.

When you create a disk group, you can set the ASM AU size to be between 1 MB and 64 MB in powers of two, such as, 1, 2, 4, 8, 16, 32, or 64. Larger AU sizes typically provide performance advantages for data warehouse applications that use large sequential reads.

About ASM Files

Files that are stored in ASM disk groups are called ASM files. Each ASM file is contained within a single ASM disk group. Oracle Database communicates with ASM in terms of files. This is identical to the way Oracle Database uses files on any file system. You can store the following file types in ASM disk groups:

- Control files
- Datafiles, temporary datafiles, and datafile copies
- SPFILEs
- Online redo logs, archive logs, and Flashback logs
- RMAN backups
- Disaster recovery configurations
- Change tracking bitmaps
- Data Pump dumpsets

Note: Oracle executables and ASCII files, such as alert logs and trace files, cannot be stored in ASM disk groups.

ASM automatically generates ASM file names as part of database operations, including tablespace creation. ASM file names begin with a plus sign (+) followed by a disk group name. You can specify user-friendly aliases for ASM files and create a hierarchical directory structure for the aliases. The following sections describe the ASM file components:

- Extents
- ASM Striping
- File Templates

Extents

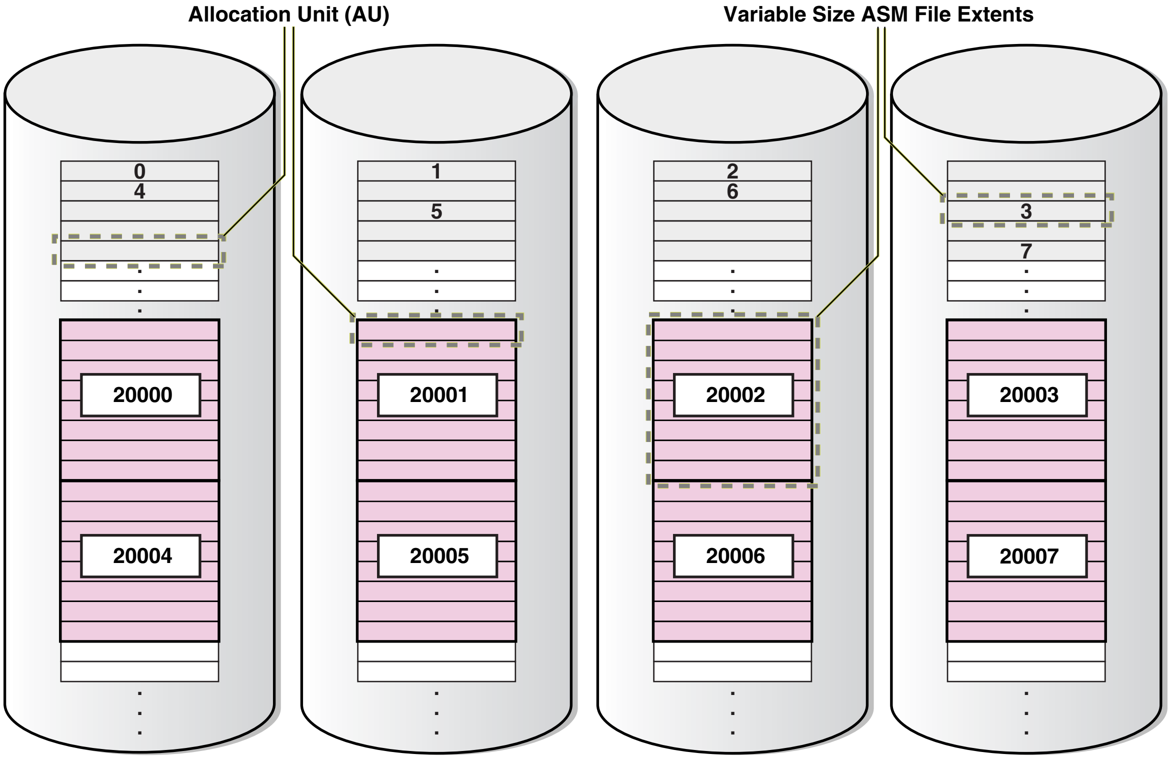
The contents of ASM files are stored in a disk group as a set, or collection, of **data extent**s that are stored on individual disks within disk groups. Each extent resides on an individual disk. Extents consist of one or more allocation units (AU). To accommodate increasingly larger files, ASM uses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

Variable size extents enable support for larger ASM datafiles, reduce SGA memory requirements for very large databases, and improve performance for file create and open operations. The size of the extent map that defines a file can be smaller by a factor of 8 and 64 depending on the file size. The initial extent size is equal to the allocation unit size and it increases by a factor of 8 and 64 at predefined thresholds. This feature is automatic for newly created and resized datafiles when the disk group compatibility attributes are set to Oracle Release 11 or higher. For information about compatibility attributes, see "Disk Group Compatibility" on page 4-21.

Figure 1–4 shows the ASM file extent relationship with allocation units. Extent size is always equal to AU for the first 20000 extent sets (0 - 19999). Figure 1–4 shows the first eight extents (0 to 7) distributed on four ASM disks. After the first 20000 extent sets, the extent size becomes 8*AU for next 20000 extent sets (20000 - 39999). This is shown as bold rectangles labeled with the extent set numbers 20000 to 20007, and so on. The next increment for an ASM extent is 64*AU (not shown in the figure).

The ASM coarse striping is always equal to the disk group AU size, but fine striping size always remains 128KB in any configuration (not shown in the figure). The AU size is determined at creation time with the allocation unit size (`AU_SIZE`) disk group attribute. The values can be 1, 2, 4, 8, 16, 32, and 64 MB.

Figure 1–4 ASM File Allocation in a Disk Group



ASM Striping

ASM striping has two primary purposes:

- To balance loads across all of the disks in a disk group
- To reduce I/O latency

Coarse-grained striping provides load balancing for disk groups while fine-grained striping reduces latency for certain file types by spreading the load more widely.

To stripe data, ASM separates files into stripes and spreads data evenly across all of the disks in a disk group. The stripes are equal in size to the effective AU. The coarse-grained stripe size is always equal to the AU size. The fine-grained stripe size always equals 128 KB; this provides lower I/O latency for small I/O operations such as redo log writes.

File Templates

Templates are collections of attribute values that are used to specify file mirroring and striping attributes for an ASM file when it is created. When creating a file, you can include a template name and assign desired attributes based on an individual file rather than the file type.

A default template is provided for every Oracle file type, but you can customize templates to meet unique requirements. Each disk group has a default template associated with each file type.

> **See Also:** Chapter 5, "Administering ASM Files, Directories, and Templates" for more information about ASM files, directories, and templates

ASM Disk Group Administration

This section describes ASM disk group administration and it contains the following topics:

- About Discovering Disks
- About Mounting Disk Groups
- About Adding and Dropping Disks

About Discovering Disks

The disk discovery process locates the operating system names for disks that ASM can access. Disk discovery is also used to find all of the disks that comprise a disk group to be mounted. This can include the disks that you want to add to a disk group and the disks that you might consider adding to a disk group.

An ASM instance requires an `ASM_DISKSTRING` initialization parameter value to specify its discovery strings. Only pathnames that the ASM instance has permission to open are discovered. The exact syntax of a discovery string depends on the platform and ASMLIB libraries. The pathnames that an operating system accepts are always usable as discovery strings.

> **See Also:** "ASM_DISKSTRING" on page 3-5 for information about the `ASM_DISKSTRING` initialization parameter, and "ASM Disk Discovery" on page 4-9 for information about disk discovery

About Mounting Disk Groups

A disk group must be mounted by a local ASM instance before database instances can access the files in the disk group. Mounting the disk group requires discovering all of the disks and locating the files in the disk group that is being mounted.

You can explicitly dismount a disk group. Oracle reports an error if you attempt to dismount a disk group when any of the disk group files are open. It is possible to have disks fail in excess of the ASM redundancy setting. If this happens, then the disk group is forcibly dismounted. This shuts down any database instances that are using the disk group.

> **See Also:** "Mounting and Dismounting Disk Groups" on page 4-27 for more information about disk groups

About Adding and Dropping Disks

The discovery string specifies the disk or disks that you can add. These disks include disks that are already in the disk group as well as new disks.

You can add a disk to an existing disk group to add space and to improve throughput. The discovery string specifies the disk or disks that you want to add. This can include disks that are already in the disk group as well as new disks. The disks that you add must be discovered by every ASM instance using the `ASM_DISKSTRING` initialization parameter. After you add a disk, ASM rebalancing operations move data onto the new disk. To minimize the rebalancing I/O, it is more efficient to add multiple disks at the same time.

You can drop a disk from a disk group if it fails or to re-provision capacity. You can also manually drop a disk that has excessive soft errors before the disk fails. Use the ASM disk name to drop a disk, not the discovery string device name. If an error occurs while writing to a disk, then Oracle drops the disk automatically.

> **See Also:** "Altering Disk Groups" on page 4-4 for more information about altering disk group membership

Online Storage Reconfigurations and Dynamic Rebalancing

Rebalancing a disk group moves data between disks to ensure that every file is evenly spread across all of the disks in a disk group. When all of the files are evenly dispersed, all of the disks are evenly filled to the same percentage; this ensures load balancing. Rebalancing does not relocate data based on I/O statistics nor is rebalancing started as a result of statistics. ASM rebalancing operations are controlled by the size of the disks in a disk group.

ASM automatically initiates a rebalance after storage configuration changes, such as when you add, drop, or resize disks. The power setting parameter determines the speed with which rebalancing operations occur.

You can manually start a rebalance to change the power setting of a running rebalance. A rebalance is automatically restarted if the instance on which the rebalancing is running stops; databases can remain operational during rebalancing operations. A rebalance has almost no effect on database performance because only one megabyte at a time is locked for relocation and only writes are blocked.

> **See Also:** "Manually Rebalancing Disk Groups" on page 4-8 for more information about disk rebalancing

2 Preparing Storage for ASM

This chapter describes how to prepare your storage subsystem before you configure Automatic Storage Management (ASM). When preparing your storage to use ASM, first determine the storage option for your system and then prepare the disk storage for the specific operating system environment as described in this chapter. This chapter contains the following topics:

Preparing Disks for ASM

You can create an ASM disk group using one of the following storage resources:

- Raw disk partition—A raw partition can be the entire disk drive or a section of a disk drive. However, the ASM disk cannot be in a partition that includes the partition table because the partition table can be overwritten.
- Logical unit numbers (LUNs)—Using hardware RAID functionality to create LUNs is a recommended approach. Storage hardware RAID 0+1 or RAID5, and other RAID configurations, can be provided to ASM as ASM disks.
- Raw logical volumes (LVM)—LVMs are supported in less complicated configurations where an LVM is mapped to a LUN, or an LVM uses disks or raw partitions. LVM configurations are not recommended by Oracle because they create a duplication of functionality. Oracle also does not recommended using LVMs for mirroring because ASM already provides mirroring.
- NFS files—ASM supports NFS files as ASM disks. Oracle Database has built-in support for the network file system (NFS) and does not depend on OS support for NFS. Although NFS and ASM have overlapping functionality, ASM can load balance or mirror across NFS files.

The procedures for preparing storage resources for ASM are:

1. Identify or create the storage devices for ASM by identifying all of the storage resource device names that you can use to create an ASM disk group. For example, on Linux systems, device names are typically presented from the `/dev` directory with the `/dev/`*`device_name_identifier`* name syntax.
2. Change the ownership and the permissions on storage device resources. For example, the following steps are required on Linux systems:

- Change the user and group ownership of devices to `oracle:dba`
- Change the device permissions to read/write
- On older Linux versions, you must configure raw device binding

After you have configured ASM, ensure that disk discovery has been configured correctly by setting the `ASM_DISKSTRING` initialization parameter.

> **Note:** Setting the ownership to `oracle:dba` is just one example that corresponds to the default settings. A non-default installation may require different settings. In general, the owner of the disk devices should be the same as the owner of the Oracle binary. The group ownership should be OSDBA of the ASM instance, which is defined at installation.

> **See Also:** "ASM_DISKSTRING" on page 3-5 for more information about the `ASM_DISKSTRING` parameter

For detailed information about preparing disks for an ASM installation, refer to your platform-specific installation guide for Oracle Database, Oracle Clusterware, and Oracle Real Application Clusters (Oracle RAC).

ASM and Multipathing

Multipathing solutions provide failover by using redundant physical path components. These components include adapters, cables, and switches that reside between the server and the storage subsystem. If one or more of these components fails, then applications can still access their data, eliminating a single point of failure with the **Storage Area Network (SAN)**, Host Bus Adapter, interface cable, or host port on a multiported storage array.

Multipathing is a software technology implemented at the operating system device driver level. Multipathing creates a pseudo device to facilitate the sharing and balancing of I/O operations across all of the available I/O paths. Multipathing also improves system performance by distributing the I/O load across all available paths. This provides a higher level of data availability through automatic failover and failback.

Although ASM is not designed with multipathing functionality, ASM does operate with multipathing technologies. Multipathing technologies are available from many sources. Storage vendors offer multipathing products to support their specific storage products, while software vendors usually develop multipathing products to support several server platforms and storage products.

> **See Also:** Your storage or software vendor multipathing documentation for more information about multipathing options for specific platforms and storage products

Using ASM with Multipathing

ASM produces an error if ASM discovers multiple disk device paths. Because a single disk can appear multiple times in a multipath configuration, you must configure ASM to discover only the multipath disk.

With ASM, you can ensure the discovery of a multipath disk by setting the value of the initialization parameter `ASM_DISKSTRING` equal to the name of the pseudo device that represents the multipath disk. For example, if you are using EMC PowerPath multipathing software, you might set `ASM_DISKSTRING` to `'/dev/rdsk/emcpower*'`. When I/O is sent to the pseudo device, the multipath driver intercepts it and provides load balancing to the underlying subpaths. When using **ASMLIB** with ASM on Linux, you can ensure the discovery of the multipath disk by configuring ASM to scan the multipath disk first or to exclude the single path disks when scanning.

See Also:

- "ASM Disk Discovery" on page 4-9 for information about disk discovery
- Articles at `https://support.oracle.com` for information about ASM and Multipathing
- Your platform-specific installation guide for information about configuring multipathing for your system

Recommendations for Storage Preparation

The following are guidelines for preparing storage for use with ASM:

- Configure two disk groups, one for the datafile and the other for the Flash Recovery Area. For availability purposes, one is used as a backup for the other.
- Ensure that LUNs, which are disk drives of partitions, that ASM disk groups use have similar storage performance and availability characteristics. In storage configurations with mixed speed drives, such as 10K and 15K RPM, I/O distribution is constrained by the slowest speed drive.
- Be aware that ASM data distribution policy is capacity-based. LUNs provided to ASM have the same capacity for each disk group to avoid an imbalance.
- Use the storage array hardware RAID 1 mirroring protection when possible to reduce the mirroring overhead on the server. Use ASM mirroring redundancy in the absence of a hardware RAID, or when you need host-based volume management functionality, such as mirroring across storage systems. You can use ASM mirroring in configurations when mirroring between geographically-separated sites over a storage interface.

  Hardware RAID 1 in some lower-cost storage products is inefficient and degrades the performance of the array. ASM redundancy delivers improved performance in lower-cost storage products.
- Maximize the number of disks in a disk group for maximum data distribution and higher I/O bandwidth.
- Create LUNs using the outside half of disk drives for higher performance. If possible, use small disks with the highest RPM.
- Create large LUNs to reduce LUN management overhead.
- Minimize I/O contention between ASM disks and other applications by dedicating disks to ASM disk groups for those disks that are not shared with other applications.
- Choose a hardware RAID stripe size that is a power of 2 and less than or equal to the size of the ASM allocation unit.
- Avoid using a Logical Volume Manager (LVM) because an LVM would be redundant. However, there are situations where certain multipathing or third party cluster solutions require an LVM. In these situations, use the LVM to

represent a single LUN without striping or mirroring to minimize the performance impact.

- For Linux, when possible, use the Oracle ASMLIB feature to address device naming and permission persistency.

  ASMLIB provides an alternative interface for the ASM-enabled kernel to discover and access block devices. ASMLIB provides storage and operating system vendors the opportunity to supply extended storage-related features. These features provide benefits such as improved performance and greater data integrity.

See Also:

- The Oracle Cloud Storage page on the Oracle Technology Network Web site at `http://www.oracle.com/technetwork/database/cloud-storage/index.html` for more information about Oracle ASM
- The Oracle ASMLib page on the Oracle Technology Network Web site at `http://www.oracle.com/technetwork/topics/linux/asmlib/index-101839.html` for information about ASMLib

Storage Considerations for Database Administrators

If you are a database administrator who is responsible for configuring your system's storage, then you need to consider not only the initial capacity of your system, but also your plans for future growth. ASM simplifies the task of accommodating growth. However, your growth plans can affect choices such as the size of the LUNs that are presented as ASM disks.

You need to also consider that I/O performance depends your **host bus adapter (HBA)** and your storage fabric, not just the storage disks. As you scale up the number of nodes in a cluster, you also need to scale up the storage subsystem.

For high availability, storage is only one component. Within storage, Oracle recommends that you configure the database work area to be separate from the recovery area. You also need a method to protect against disk failures by using hardware mirroring or host-based mirroring from a normal or high redundancy disk group. Furthermore, you also need to consider multipathing for HBAs and the fabric when considering storage availability. With ASM mirroring, the failure group configuration also affects high availability.

3 Administering ASM Instances

This chapter describes how to administer Automatic Storage Management (ASM) instances. It explains how to configure ASM instance parameters as well how to set Oracle Database parameters for use with ASM. The chapter also describes ASM instance administration as well as upgrading, patching, and authentication for ASM instance access. You can also use procedures in this chapter to migrate a database to use ASM.

Administering an ASM instance is similar to administering an Oracle Database instance, but the process requires fewer procedures. You can use Oracle Enterprise Manager and SQL*Plus to perform ASM instance administration tasks. This chapter contains the following topics:

> **See Also:** "About ASM Instances" on page 1-2 for a description of an ASM instance and Chapter 6, "Administering ASM with Oracle Enterprise Manager" for information about using Enterprise Manager to administer ASM

Operating With Different Releases of ASM and Database Instances Simultaneously

Automatic Storage Management (ASM) in Oracle Database 11*g* supports both older and newer software versions of Oracle database instances, including Oracle Database 10*g*. Both forward and backward compatibility is maintained between Oracle Database 10*g* and 11*g*, enabling combinations of 10.1, 10.2 and 11.1 releases for ASM and database instances to successfully interoperate. For compatibility between Oracle Clusterware and ASM, the Oracle Clusterware release must be greater than or equal to the ASM release.

There are additional compatibility considerations when using disk groups with different releases of ASM and database instances. For information about disk group compatibility attributes settings, see "Disk Group Compatibility" on page 4-21.

When using different software versions, the database instance supports ASM functionality of the earliest release in use. For example:

- A 10.1 database instance operating with an 11.1 ASM instance supports only ASM 10.1 features.
- An 11.1 database instance operating with a 10.1 ASM instance supports only ASM 10.1 features.

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

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

You can query the `V$ASM_CLIENT` view on both ASM and database instances. For an example showing a query on the `V$ASM_CLIENT` view, see Example 4–4 on page 4-31. For more information about the `V$ASM_CLIENT` and `V$ASM_*` views, see "Using Views to Obtain ASM Information" on page 4-29.

Configuring Initialization Parameters for an ASM Instance

This section discusses initialization parameter files and parameter settings for ASM instances. To install and initially configure an ASM instance, use Oracle Universal Installer (OUI) and Database Configuration Assistant (DBCA). Refer to your platform-specific *Oracle Database Installation Guide* for details about installing and configuring ASM.

After an ASM instance has been installed on a single-instance Oracle Database or in an Oracle Real Application Clusters (Oracle RAC) environment, the final ASM configuration can be performed. You only need to configure a few ASM-specific instance initialization parameters. The default values are sufficient in most cases.

> **See Also:** The Oracle Cloud Storage page on the Oracle Technology Network Web site at `http://www.oracle.com/technetwork/database/cloud-storage/index.html` for more information about Oracle ASM best practices

This section contains the following topics:

- Initialization Parameter Files for an ASM Instance
- Setting ASM Initialization Parameters
- ASM Parameter Setting Recommendations
- Setting Database Initialization Parameters for Use with ASM
- Disk Group Attributes

> **See Also:**
>
> - *Oracle Database Reference* for information about initialization parameters
> - *Oracle Database Administrator's Guide* for information about initialization parameter files

Initialization Parameter Files for an ASM Instance

When installing ASM for a single-instance Oracle Database, DBCA creates a separate server parameter file (SPFILE) and password file for the ASM instance. When installing ASM in a clustered ASM environment where the ASM home is shared among all of the nodes, DBCA creates an SPFILE for ASM. In a clustered environment without a shared ASM home, DBCA creates a text-based initialization parameter file (PFILE) for ASM on each node.

You can use an SPFILE or PFILE as the ASM instance parameter file. If you use an SPFILE in a clustered ASM environment, then you must place the SPFILE on a shared raw device or on a cluster file system. If you do not use a shared ASM home, then the ASM instance uses a PFILE.

The same rules for file name, default location, and search order that apply to database initialization parameter files also apply to ASM initialization parameter files. For example, in single-instance UNIX and Linux Oracle Database environments, the server parameter file for ASM has the following path:

```
$ORACLE_HOME/dbs/spfile+ASM.ora
```

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for more information about creating and maintaining an initialization parameter files
> - *Oracle Database 2 Day DBA* for information about viewing and modifying initialization parameters

Setting ASM Initialization Parameters

There are several initialization parameters that you must set for an ASM instance. You can set these parameters when you create your database using DBCA. You can also set some of these parameters after database creation using Oracle Enterprise Manager or SQL `ALTER SYSTEM` or `ALTER SESSION` statements.

The `INSTANCE_TYPE` initialization parameter is the only required parameter in the ASM instance parameter file. The ASM* parameters use suitable defaults for most environments. You cannot use parameters with names that are prefixed with ASM* in database instance parameter files.

Some database initialization parameters are also valid for an ASM instance initialization file. In general, ASM selects the appropriate defaults for database parameters that are relevant to an ASM instance.

> **See Also:** "Configuring ASM Initialization Parameters" on page 6-3 for information about setting `ASM*` parameters with Oracle Enterprise Manager

Automatic Memory Management for ASM

Automatic memory management automatically manages the memory-related parameters for both ASM and database instances with the `MEMORY_TARGET` parameter. Automatic memory management is enabled by default on an ASM instance, 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. Oracle strongly recommends that you use automatic memory management for ASM.

If you do not set a value for `MEMORY_TARGET`, but you do set values for other memory related parameters, Oracle internally calculates the optimum value for `MEMORY_TARGET` based on those memory parameter values. You can also increase MEMORY_TARGET dynamically, up to the value of the MEMORY_MAX_TARGET parameter, just as you can do for the database instance.

Although it is not recommended, you can disable automatic memory management by either setting the value for MEMORY_TARGET to 0 in the ASM parameter file or by running an ALTER SYSTEM SET MEMORY_TARGET=0 statement. When you disable automatic memory management, Oracle reverts to auto shared memory management and automatic PGA memory management. If you want to revert to Oracle Database 10*g* release 2 (10.2) functionality to manually manage ASM SGA memory, also run the ALTER SYSTEM SET SGA_TARGET=0 statement. You can then manually manage ASM memory using the information in "ASM Parameter Setting Recommendations" on page 3-4, that discusses ASM memory-based parameter settings. Unless specified, the behaviors of all of the automatic memory management parameters in ASM instances is the same as in Oracle Database instances.

Note: For a Linux environment, automatic memory management cannot work if /dev/shm is not available or is undersized. For more information, see *Oracle Database Administrator's Reference for Linux and UNIX*. For information about platforms that support automatic memory management, see *Oracle Database Administrator's Guide*.

Note: The minimum MEMORY_TARGET for ASM is 256 MB. If you set MEMORY_TARGET to 100 MB, then Oracle increases the value for MEMORY_TARGET to 256 MB automatically.

See Also:

- *Oracle Database Administrator's Guide* for more information about the functionality of automatic memory management for database instances, which varies from ASM
- *Oracle Database Concepts* for an overview of memory management methods

ASM Parameter Setting Recommendations

This section contains information about the following parameters for ASM:

- ASM_DISKGROUPS
- ASM_DISKSTRING
- ASM_POWER_LIMIT
- ASM_PREFERRED_READ_FAILURE_GROUPS
- DB_CACHE_SIZE
- DIAGNOSTIC_DEST
- INSTANCE_TYPE
- LARGE_POOL_SIZE
- PROCESSES
- REMOTE_LOGIN_PASSWORDFILE

- SHARED_POOL_SIZE

See Also:

- *Oracle Database Administrator's Guide* for more information about creating and maintaining an initialization parameter file
- *Oracle Database 2 Day DBA* for information about viewing and modifying initialization parameters

ASM_DISKGROUPS

The ASM_DISKGROUPS initialization parameter specifies a list of the names of disk groups that an ASM instance mounts at startup. Oracle ignores the value that you set for ASM_DISKGROUPS when you specify the NOMOUNT option at startup or when you issue the ALTER DISKGROUP ALL MOUNT statement. The default value of the ASM_DISKGROUPS parameter is a NULL string. If the parameter value is NULL or is not specified, then ASM does not mount any disk groups.

The ASM_DISKGROUPS parameter is dynamic. If you are using a server parameter file (SPFILE), then you should not need to manually alter the value of ASM_DISKGROUPS. ASM automatically adds a disk group to this parameter when the disk group is successfully created or mounted. ASM also automatically removes a disk group from this parameter when the disk group is dropped or dismounted. The following is an example of setting the ASM_DISKGROUPS parameter dynamically:

```
SQL> ALTER SYSTEM SET ASM_DISKGROUPS = 'CONTROLFILE, DATAFILE, LOGFILE, STANDBY'
```

When using a text initialization parameter file (PFILE), you must edit the initialization parameter file to add the name of any disk group that you want mounted automatically at instance startup. You must remove the name of any disk group that you no longer want automatically mounted. The following is an example of the ASM_DISKGROUPS parameter in the initialization file:

```
ASM_DISKGROUPS = CONTROLFILE, DATAFILE, LOGFILE, STANDBY
```

Note: Issuing the ALTER DISKGROUP...ALL MOUNT or ALTER DISKGROUP...ALL DISMOUNT commands does not affect the value of ASM_DISKGROUPS.

See Also: "Mounting and Dismounting Disk Groups" on page 4-27 for additional information about ASM disk groups and *Oracle Database Reference* for more information about the ASM_DISKGROUPS initialization parameter

ASM_DISKSTRING

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

The discovery string format depends on the 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/ directory, set ASM_DISKSTRING to:

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

The asterisk is required. To limit the discovery process to only include disks that have a name that ends in disk3 or disk4, set ASM_DISKSTRING to:

```
/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 ASM to search a default path for all disks in the system to which the 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.

ASM cannot use a disk unless all of the ASM instances in the cluster can discover the disk through one of their own discovery strings. The names do not need 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.

> **See Also:** *Oracle Database Reference* for more information about the ASM_DISKSTRING initialization parameter

ASM_POWER_LIMIT

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 11 inclusive. A value of 0 disables rebalancing. Higher numeric values enable the rebalancing operation to complete more quickly, but might result in higher I/O overhead.

> **See Also:** "Tuning Rebalance Operations" on page 4-9 for more information about ASM_POWER_LIMIT and *Oracle Database Reference* for more information about the ASM_POWER_LIMIT initialization parameter

ASM_PREFERRED_READ_FAILURE_GROUPS

The ASM_PREFERRED_READ_FAILURE_GROUPS initialization parameter value is a comma-delimited list of strings that specifies the failure groups that should be preferentially read by the given instance. This parameter is generally used only for clustered ASM instances and its value can be different on different nodes. For example:

diskgroup_name1.failure_group_name1, ...

The ASM_PREFERRED_READ_FAILURE_GROUPS parameter setting is instance specific. This parameter is only valid for clustered ASM instances and the default value is NULL.

> **Note:** The ASM_PREFERRED_READ_FAILURE_GROUPS parameter is valid only in Oracle RAC environments.

See Also:

- "Preferred Read Failure Groups" on page 4-19 for more information about ASM_PREFERRED_READ_FAILURE_GROUPS
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring preferred disks in extended clusters
- *Oracle Database Reference* for more information about the ASM_PREFERRED_READ_FAILURE_DISKGROUPS initialization parameter

DB_CACHE_SIZE

You do not need to set a value for the DB_CACHE_SIZE initialization parameter if you use automatic memory management.

The setting for the DB_CACHE_SIZE parameter determines the size of the buffer cache. This buffer cache is used to store metadata blocks. The default value for this parameter is suitable for most environments.

See Also:

- *Oracle Database Administrator's Guide* for more information about setting the DB_CACHE_SIZE initialization parameter
- *Oracle Database Performance Tuning Guide* for more information about memory configuration
- *Oracle Database Reference* for more information about the DB_CACHE_SIZE parameter

DIAGNOSTIC_DEST

The DIAGNOSTIC_DEST initialization parameter specifies the directory where diagnostics for an instance are located. The value for an ASM instance is of the form:

```
diagnostic_dest/diag/asm/db_name/instance_name
```

For an ASM instance, *db_name* defaults to +asm.

See Also:

- *Oracle Database Administrator's Guide* for more information about setting the DIAGNOSTIC_DEST initialization parameter
- *Oracle Database Reference* for more information about the DIAGNOSTIC_DEST parameter

INSTANCE_TYPE

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
```

See Also: *Oracle Database Reference* for more information about the INSTANCE_TYPE parameter

LARGE_POOL_SIZE

You do not need to set a value for the LARGE_POOL_SIZE initialization parameter if you use automatic memory management.

The setting for the LARGE_POOL_SIZE parameter is used for large allocations. The default value for this parameter is suitable for most environments.

See Also:

- *Oracle Database Administrator's Guide* for more information about setting the LARGE_POOL_SIZE initialization parameter
- *Oracle Database Performance Tuning Guide* for more information about memory configuration
- *Oracle Database Reference* for more information about the LARGE_POOL_SIZE parameter

PROCESSES

You do not need to set a value for the PROCESSES initialization parameter if you use automatic memory management.

The PROCESSES initialization parameter affects ASM, but generally you do not need to modify the setting. The default value provided is usually suitable.

See Also:

- *Oracle Database Administrator's Guide* for more information about setting the PROCESSES initialization parameter
- *Oracle Database Reference* for more information about the PROCESSES parameter

REMOTE_LOGIN_PASSWORDFILE

The REMOTE_LOGIN_PASSWORDFILE initialization parameter specifies whether the ASM instance checks for a password file. This parameter operates the same for ASM and database instances.

See Also:

- *Oracle Database Administrator's Guide* for more information about setting the REMOTE_LOGIN_PASSWORDFILE initialization parameter
- *Oracle Database Reference* for more information about the REMOTE_LOGIN_PASSWORDFILE parameter

SHARED_POOL_SIZE

You do not need to set a value for the SHARED_POOL_SIZE initialization parameter if you use automatic memory management.

The setting for the SHARED_POOL_SIZE parameter determines the amount of memory required to manage the instance. The setting for this parameter is also used to determine the amount of space that is allocated for extent storage. The default value for this parameter is suitable for most environments.

See Also:

- *Oracle Database Administrator's Guide* for more information about setting the SHARED_POOL_SIZE initialization parameter
- *Oracle Database Performance Tuning Guide* for more information about memory configuration
- *Oracle Database Reference* for more information about the SHARED_POOL_SIZE parameter

Setting Database Initialization Parameters for Use with ASM

When you do not use automatic memory management in a database instance, the SGA parameter settings for a database instance may require minor modifications to support ASM. When you use automatic memory management, the sizing data discussed in this section can be treated as informational only or as supplemental information to help determine the appropriate values that you should use for the SGA. Oracle highly recommends using automatic memory management.

See Also:

- *Oracle Database Administrator's Guide* for information about managing memory allocation in an Oracle Database instance
- *Oracle Database Performance Tuning Guide* for more information about memory configuration and use

The following are guidelines for SGA sizing on the database instance:

- PROCESSES initialization parameter—Add 16 to the current value
- LARGE_POOL_SIZE initialization parameter—Add an additional 600K to the current value
- SHARED_POOL_SIZE initialization parameter—Aggregate the values from the following queries to obtain the current database storage size that is either already on ASM or will be stored in ASM. Next, determine the redundancy type and calculate the SHARED_POOL_SIZE using the aggregated value as input.

```
SELECT SUM(bytes)/(1024*1024*1024) FROM V$DATAFILE;
SELECT SUM(bytes)/(1024*1024*1024) FROM V$LOGFILE a, V$LOG b
       WHERE a.group#=b.group#;
SELECT SUM(bytes)/(1024*1024*1024) FROM V$TEMPFILE
       WHERE status='ONLINE';
```

  - For disk groups using external redundancy, every 100 GB of space needs 1 MB of extra shared pool plus 2 MB
  - For disk groups using normal redundancy, every 50 GB of space needs 1 MB of extra shared pool plus 4 MB
  - For disk groups using high redundancy, every 33 GB of space needs 1 MB of extra shared pool plus 6 MB

See Also:

- *Oracle Database Administrator's Guide* for information about managing memory allocation in an Oracle Database instance
- *Oracle Database Performance Tuning Guide* for more information about memory configuration and use

Disk Group Attributes

Disk group attributes are essentially parameters that are bound to a disk group, rather than an instance. The disk group attributes are:

- AU_SIZE

  For information about allocation unit size and extents, see "Extents" on page 1-7. For an example of the use of the AU_SIZE attribute, see Example 4–1, "Creating a Disk Group" on page 4-3.

- COMPATIBLE.ASM

  For information about the COMPATIBLE.ASM attribute, see "COMPATIBLE.ASM" on page 4-22.

- COMPATIBLE.RDBMS

  For information about the COMPATIBLE.RDBMS attribute, see "COMPATIBLE.RDBMS" on page 4-22.

- DISK_REPAIR_TIME

  For information about the DISK_REPAIR_TIME attribute, see "ASM Fast Mirror Resync" on page 4-17.

Administering ASM Instances

The following section describes how to administer ASM instances under the following topics:

- Administering ASM Instances with Server Control Utility
- Starting Up an ASM Instance
- Shutting Down an ASM Instance
- ASM Background Processes

Administering ASM Instances with Server Control Utility

In addition to the ASM administration procedures that this section describes, you can use Server Control Utility (SRVCTL) in clustered ASM environments to perform the following ASM administration tasks:

- Add and remove ASM instance records in the Oracle Cluster Registry (OCR)
- Enable, disable, start, and stop ASM instances
- Display the ASM instance configuration and status

> **See Also:** The *Oracle Real Application Clusters Administration and Deployment Guide* for detailed information about administering ASM instances with SRVCTL

Starting Up an ASM Instance

You start an ASM instance similarly to the way in which you start an Oracle database instance with some minor differences. 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. Depending on your operating system and whether you installed ASM in a separate ASM home, you might have to change other environment variables.

- 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. If you have not entered a value for ASM_DISKGROUPS, then the ASM instance starts and Oracle displays an error that no disk groups were mounted. You can then mount disk groups with the ALTER DISKGROUP...MOUNT command.

  > **See Also:** "Mounting and Dismounting Disk Groups" on page 4-27 for more information

  ASM provides a MOUNT FORCE option to enable ASM disk groups to be mounted in normal or high redundancy modes even though some ASM disks may be unavailable to the disk group at mount time. The default behavior without the FORCE option is to fail to mount a disk group that has damaged or missing disks.

  To successfully mount with the MOUNT FORCE option, ASM must be able to find at least one copy of the extents for all of the files in the disk group. In this case, ASM can successfully mount the disk group, but with potentially reduced redundancy. If all disks are available, then using the FORCE option causes the MOUNT command to fail as well. This discourages unnecessary and improper use of the feature.

  ASM puts the unavailable disks in an offline mode if ASM is unable to access them. ASM then begins timing the period that these disks are in an offline mode. If the disk offline time period exceeds the timer threshold, then ASM permanently drops those disks from the disk group. You can change the offline timer after a disk is put in an offline state by using the ALTER DISKGROUP OFFLINE statement.

  The MOUNT FORCE option is useful in situations where a disk is temporarily unavailable and you want to mount the disk group with reduced redundancy while you correct the situation that caused the outage.

  > **Note:** An ASM instance mounts an incomplete disk group differently depending on the specified compatibility as discussed under the heading "Disk Group Compatibility" on page 4-21.

- The associated Oracle database instance does not need to be running when you start the associated ASM instance.

The following list describes how ASM interprets SQL*Plus STARTUP command parameters.

- FORCE Parameter

  Issues a SHUTDOWN ABORT to the ASM instance before restarting it.

- MOUNT or OPEN Parameter

  Mounts the disk groups specified in the ASM_DISKGROUPS initialization parameter. This is the default if no command parameter is specified.

- NOMOUNT Parameter

  Starts up the ASM instance without mounting any disk groups.

- RESTRICT Parameter

  Starts up an instance in restricted mode that enables access only to users with both the CREATE SESSION and RESTRICTED SESSION system privileges. The RESTRICT clause can be used in combination with the MOUNT, NOMOUNT, and OPEN clauses.

  > **See Also:** "About Restricted Mode" on page 3-12 for more information

  In restricted mode, database instances cannot use the disk groups. In other words, databases cannot open files that are in that disk group. Also, the disk group cannot be mounted by any other instance in the cluster. Mounting the disk group in restricted mode enables only one ASM instance to mount the disk group. This mode is useful to mount the disk group for repairing configuration issues.

The following is a sample SQL*Plus session for starting an ASM instance.

```
SQLPLUS /NOLOG
SQL> CONNECT SYS AS SYSASM
Enter password: sys_password
Connected to an idle instance.

SQL> STARTUP
ASM instance started

Total System Global Area   71303168 bytes
Fixed Size                 1069292 bytes
Variable Size              45068052 bytes
ASM Cache                  25165824 bytes
ASM disk groups mounted
```

> **See Also:** "Authentication for Accessing ASM Instances" on page 3-17 for more information about user authentication

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for more information about using environment variables to select instances
> - *Oracle Database Administrator's Guide* for more information about starting up and shutting down Oracle instances
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information about starting an ASM instance with SRVCTL in Oracle RAC

About Restricted Mode

You can use the STARTUP RESTRICT command to control access to an ASM instance while you perform maintenance. When an ASM instance is active in this mode, all of the disk groups that are defined in the ASM_DISKGROUPS parameter are mounted in RESTRICTED mode. This prevents databases from connecting to the ASM instance. In addition, the restricted clause of the ALTER SYSTEM statement is disabled for the ASM instance. The ALTER DISKGROUP *diskgroupname* MOUNT statement is extended to enable ASM to mount a disk group in restricted mode.

When you mount a disk group in RESTRICTED mode, the disk group can only be mounted by one instance. Clients of ASM on that node cannot access that disk group while the disk group is mounted in RESTRICTED mode. The RESTRICTED mode enables you to perform maintenance tasks on a disk group in the ASM instance 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 an Oracle RAC 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.

Cluster Synchronization Services Requirements for ASM

The Cluster Synchronization Services (CSS) daemon provides cluster services for ASM, communication between the ASM and database instances, and other essential services. When DBCA creates a database, the CSS daemon is usually started and configured to start upon restart. If DBCA created the database, then you must ensure that the CSS daemon is running before you start the ASM instance.

CSS Daemon on UNIX and Linux Computers To determine if the CSS daemon is running, run the command `crsctl check cssd`. If Oracle displays the message `CSS appears healthy`, then the CSS daemon is running. Otherwise, to start the CSS daemon and configure the host to always start the daemon upon restart, do the following:

1. Log in to the host as the `root` user.
2. Ensure that the entry `$ORACLE_HOME/bin` is in your `PATH` environment variable.
3. Run the following command:

   ```
   localconfig add
   ```

CSS Daemon on Microsoft Windows Computers You can also use the `crsctl` and `localconfig` commands to check the status of the CSS daemon or to start it. To use Windows GUI tools to determine whether the CSS daemon is properly configured and running, double-click the **Services** icon in the Windows Control Panel and locate the **OracleCSService** service. The service's status should be `Started` and its startup type should be `Automatic`.

Note: Refer to your Windows documentation for information about how to start a Windows service and how to configure it for automatic startup.

Shutting Down an ASM Instance

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` environment variable is set to the ASM SID so that you can connect to the ASM instance. Depending on your operating system and whether you installed ASM in a separate ASM home, you might have to change other environment variables before starting SQL*Plus. Oracle strongly recommends that you shut down all database instances that use the ASM instance before attempting to shut down the ASM instance.

See Also:

- *Oracle Database Administrator's Guide* for more information about using environment variables to select instances
- *Oracle Database Administrator's Guide* for more information about starting up and shutting down Oracle instances

```
SQLPLUS /NOLOG
SQL> CONNECT SYS AS SYSASM
Enter password: sys_password
Connected.
SQL> SHUTDOWN NORMAL
```

See Also: "Authentication for Accessing ASM Instances" on page 3-17 for more information about use authentication

The following list describes the SHUTDOWN modes and describes the behavior of the ASM instance in each mode.

- NORMAL Clause

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

- IMMEDIATE or TRANSACTIONAL Clause

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

- ABORT Clause

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

See Also: "Authentication for Accessing ASM Instances" on page 3-17 for more information about user authentication on ASM instance

ASM Background Processes

The following background processes are an integral part of Automatic Storage Management:

- ARB*n* performs the actual rebalance data extent movements in an Automatic Storage Management instance. There can be many of these processes running at a time, named ARB0, ARB1, and so on.

- ASMB runs in a database instance that is using an ASM disk group. ASMB communicates with the ASM instance, managing storage and providing statistics. ASMB can also run in the ASM instance. ASMB runs in ASM instances when the ASMCMD `cp` command runs or when the database instance first starts if the SPFILE is stored in ASM.
- GMON maintains disk membership in ASM disk groups.
- MARK marks ASM allocation units as stale following a missed write to an offline disk. This essentially tracks which extents require resync for offline disks.
- RBAL runs in both database and ASM instances. In the database instance, it does a global open of ASM disks. In an ASM instance, it also coordinates rebalance activity for disk groups.

The processes described in the previous list are important for the ASM instance and should not be modified. In addition to the processes listed in this section, there are additional processes that run in both the ASM and database instances, such as database writer process (DBW*n*), log writer process (LGWR), Process Monitor Process (PMON), and System Monitor Process (SMON).

Also, there are ASM slave processes that run periodically to perform a specific task. For example, the S*nnn* transient slave process is responsible for performing the resync of extents at the time that the disk is brought online. The slave processes are not technically background processes.

For more information about Oracle database background processes, see the discussion about background processes in *Oracle Database Concepts*. For a description of the `V$BGPROCESS` view that displays information about background processes, see the *Oracle Database Reference*.

Using ASM Rolling Upgrades

ASM rolling upgrades enable you to independently upgrade or patch clustered ASM nodes without affecting database availability, thus providing greater uptime. Rolling upgrade means that all of the features of a clustered ASM environment function when one or more of the nodes in the cluster uses different software versions.

> **Note:** Rolling upgrades only apply to clustered ASM instances, and you can only perform rolling upgrades on environments with Oracle Database 11*g* or later. In other words, you cannot use this feature to upgrade from Oracle Database 10*g* to Oracle Database 11*g*.

To perform a rolling upgrade, your environment must be prepared. If you are using Oracle Clusterware, then your Oracle Clusterware must be fully upgraded to the next patch or release version before you start the ASM rolling upgrade. In addition, you should prepare your Oracle Clusterware in a rolling upgrade manner to ensure high availability and maximum uptime.

Before you patch or upgrade the ASM software on a node, you must place the ASM cluster into rolling upgrade mode. This enables you to begin an upgrade and operate your environment in multiversion software mode. Do this by issuing the following SQL statement where *`number`* includes the version number, release number, update number, port release number, and port update number. Enter these values for *`number`* in a decimal-separated string enclosed in single quotation marks, for example, `'11.1.0.7.0'`, to perform the upgrade as in the following example

```
ALTER SYSTEM START ROLLING MIGRATION TO '11.1.0.7.0';
```

The instance from which you run this statement verifies whether the value that you specified for number is compatible with the current installed version of your software. When the upgrade begins, the behavior of the clustered ASM environment changes, and only the following operations are permitted on the ASM instance:

- Disk group mount and dismount
- Database file open, close, resize, and delete
- Limited access to fixed views and fixed packages

> **Note:** You can query fixed views and run anonymous PL/SQL blocks using fixed packages, such as `DBMS_DISKGROUP`. However, only local views are available; Oracle disables all global views when a clustered ASM environment is in rolling upgrade mode.

After the rolling upgrade has been started, you can shut down each ASM instance and perform the software upgrade. On start up, the updated ASM instance can rejoin the cluster. When you have migrated all of the nodes in your clustered ASM environment to the latest software version, you can end the rolling upgrade mode.

If a disk goes offline when the ASM instance is in rolling upgrade mode, then the disk remains offline until the rolling upgrade has ended. Also, the timer for dropping the disk is stopped until the ASM cluster is out of rolling upgrade mode.

> **See Also:** "ASM Fast Mirror Resync" on page 4-17 for more information about using fast mirror resync during a rolling upgrade

You can also use the same procedure to roll back node upgrades if you encounter problems with the upgrade. The ASM functionality is compatible with the lowest software version that is on any of the nodes in the cluster during an upgrade.

The upgrade fails if there are rebalancing operations occurring anywhere in the cluster. You must wait until the rebalance completes before attempting to start a rolling upgrade. In addition, as long as there is one instance active in the cluster, the rolling upgrade state is preserved.

New instances that join the cluster immediately switch to a rolling upgrade state on startup. In other words, if a rolling upgrade is in progress in a clustered ASM environment and if any new ASM instance joins the cluster, then the new ASM instance is notified that the cluster is in rolling upgrade mode. You can use the following SQL function to query the state of a clustered ASM environment:

```
SELECT SYS_CONTEXT('sys_cluster_properties', 'cluster_state') FROM DUAL;
```

If all of the instances in a clustered ASM environment stop running, then when any of the ASM instances restart, the restarted instance will not be in rolling upgrade mode. To perform the upgrade after your instances restart, you must re-run the commands to restart the rolling upgrade operation. When the rolling upgrade completes, run the following SQL statement:

```
ALTER SYSTEM STOP ROLLING MIGRATION;
```

After you run this statement, Oracle performs the following operations:

- Validates that all of the members of the cluster are at the same software version. If there are one or more ASM instances that have different versions, then Oracle displays an error and the cluster continues to be in rolling upgrade mode.

- Updates the cluster-wide state so that the ASM instances are no longer in rolling upgrade mode; the ASM instances begin supporting the full clustered ASM functionality.
- Rebalance operations that were pending are restarted if the setting for the `ASM_POWER_LIMIT` parameter enables this.

> **See Also:** "ASM_POWER_LIMIT" on page 3-6 for more information about using the `ASM_POWER_LIMIT` parameter

Patching ASM Instances

For Oracle RAC environments, if you configure ASM in a home that is separate from the Oracle Database home, then when you apply patches you must apply them in a specific order. You must first ensure that your Oracle Clusterware version is at least equal to the version of the patch that you are applying. This may require you to patch the Oracle Clusterware home first. Then apply the patch to the ASM home, and finally, apply the patch to the Oracle Database home.

> **Note:** You must apply the patch to the ASM home before you apply it to the Oracle Database home.

Authentication for Accessing ASM Instances

This section describes the following topics:

- About the SYSASM Privilege for ASM
- Accessing an ASM Instance
- Creating Users with the SYSASM Privilege
- Operating System Authentication for ASM
- Password File Authentication for ASM

The ASM and database instances must have equivalent operating system access rights. For example, the ASM instance and the database instance must have identical read and write permissions for the disks that comprise the related ASM disk group. For UNIX systems, this is typically provided through shared UNIX group membership. On Windows systems, the ASM service can run as Administrator.

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`. There are three modes of connecting to ASM instances:

- Local connection using operating system authentication
- Local connection using password authentication
- Remote connection by way of Oracle Net Services using password authentication

> **Note:** If you create an ASM instance using Database Configuration Assistant (DBCA), or if you create the ASM instance using Database Upgrade Assistant (DBUA), then the user `SYS` should have SYSASM privileges.

> **See Also:** Your operating system-specific Oracle Database installation guide for information about how to ensure that the ASM and database instances have member disk access

About the SYSASM Privilege for ASM

SYSASM is a system privilege that enables the separation of the SYSDBA database administration privilege from the ASM storage administration privilege. Access to the SYSASM privilege is granted by membership in an operating system group that is designated as the OSASM group. This is similar to SYSDBA and SYSOPER privileges, which are system privileges granted through membership in the groups designated as the OSDBA and OSOPER operating system groups. You can designate one group for all of these system privileges, or you can designate separate groups for each operating system privilege.

You can divide system privileges during ASM installation, so that database administrators, storage administrators, and database operators each have distinct operating system privilege groups. Use the Custom Installation option to designate separate operating system groups as the operating system authentication groups for privileges on ASM. Table 3–1 lists the operating system authentication groups for ASM, and the privileges that their members are granted:

Table 3–1 Operating System Authentication Groups for ASM

Group	Privilege Granted to Members
OSASM	SYSASM privilege, which provides full administrative privilege for the ASM instance.
OSDBA for ASM	SYSDBA privilege on the ASM instance. This privilege grants access to data stored on ASM, and in the current release, grants the SYSASM administrative privileges.
OSOPER for ASM	SYSOPER privilege on the ASM instance.

If you do not want to divide system privileges access into separate operating system groups, then you can designate one operating system group as the group whose members are granted access as OSDBA, OSOPER, OSASM, and OSDBA for ASM, and OSOPER for ASM privileges. The default operating system group name for all of these is `dba`. You can also specify OSASM, OSDBA for ASM, and OSOPER for ASM when you perform a custom installation of ASM. Furthermore, you can specify OSDBA and OSOPER when performing a custom database installation.

Whether you create separate operating system privilege groups or use one group to provide operating system authentication for all system privileges, you should use SYSASM to connect to and administer an ASM instance. In Oracle 11*g* release 1, both SYSASM and SYSDBA are supported privileges; however, if you use the SYSDBA privilege to administer an ASM instance, then Oracle will write warning messages to the alert log, indicating that the SYSDBA privilege is deprecated on an ASM instance for administrative commands. In a future release, the privilege to administer an ASM instance with SYSDBA will be removed.

Operating system authentication using membership in the group or groups designated as OSDBA, OSOPER, and OSASM is valid on all Oracle platforms. Connecting to an ASM instance as SYSASM grants you full access to all of the available ASM disk groups and management functions.

Accessing an ASM Instance

This section describes how to connect to an ASM instance. In the examples where you provide a user name, you are prompted for a password.

> **Note:** The SYS user is created by default by DBCA during installation process with all three system privileges.

Use the following statement to connect locally to an ASM instance using operating system authentication:

```
sqlplus / AS SYSASM
```

Use the following statement to connect locally using password authentication:

```
sqlplus SYS AS SYSASM
```

Use the following statement to connect remotely using password authentication:

```
sqlplus sys@\"myhost.mydomain.com:1521/asm\" AS SYSASM
```

Use the following statement to connect to an ASM instance with SYSDBA privilege:

```
sqlplus / AS SYSDBA
```

Oracle writes messages to the alert log if you issue ASM administrative commands that will be accessible only to the SYSASM privilege in future releases.

Creating Users with the SYSASM Privilege

When you are logged in to an ASM instance as SYSASM, you can use the combination of CREATE USER and GRANT SQL statements to create a new user who has the SYSASM privilege. These commands update the password file for the local ASM instance. Similarly, you can revoke the SYSASM privilege from a user using the REVOKE command, and you can drop a user from the password file using the DROP USER command. The following example describes how to perform these operations for the user identified as *new_user*:

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

REM connect the user to the ASM instance
SQL> CONNECT new_user AS SYSASM;
Enter password:

REM revoke the SYSASM privilege, then drop the user
SQL> REVOKE SYSASM FROM new_user;
SQL> DROP USER new_user;
```

Operating System Authentication for ASM

Membership in the operating system group designated as the OSASM group provides operating system authentication for the SYSASM system privilege. OSASM is provided exclusively for ASM. Initially, only the user that installs ASM is a member of the OSASM group, if you use a separate operating system group for that privilege. However, you can add other users. Members of the OSASM group are authorized to connect using the SYSASM privilege and have full access to ASM, including administrative access to all disk groups that are managed by that ASM instance.

On Linux and UNIX systems, the default operating system group designated as OSASM, OSOPER, and OSDBA is dba. On Windows systems, the default name designated as OSASM, OSOPER, and OSDBA is ora_dba.

Note: The user who is the software owner for the Oracle Database home, that Oracle documentation describes as the oracle user must be a member of the group that is designated as the OSDBA group for the ASM home. This is automatically configured when ASM and an Oracle Database share the same Oracle home. If you install the ASM and database instances in separate homes, then you must ensure that you create a separate OSDBA group for ASM, and that you designate the correct group memberships for each OSDBA group. Otherwise, the database instance will not be able to connect to the ASM instance.

See Also: *Oracle Database Administrator's Guide* for more information about using operating system authentication

Password File Authentication for ASM

Password file authentication for ASM can work both locally and remotely. To enable password file authentication, you must create a password file for ASM. A password file is also required to enable Oracle Enterprise Manager to connect to ASM remotely.

If you select the ASM storage option, then DBCA creates a password file for ASM when it initially configures the ASM disk groups. Similar to a database password file, the only user added to the password file when DBCA creates it is SYS. To add other users to the password file, you can use the CREATE USER and GRANT commands as described previously in the section titled "About the SYSASM Privilege for ASM" on page 3-18.

If you configure an ASM instance without using DBCA, then you must manually create a password file and GRANT the SYSASM privilege to user SYS.

See Also:

- *Oracle Database Administrator's Guide* for information about creating and maintaining a password file
- *Oracle Database SQL Language Reference* for information about the `CREATE USER` and `GRANT` commands
- *Oracle Database Security Guide* for information about database security
- *Oracle Database Reference* for information about the `V$PWFILE_USERS` view which lists users who have been granted `SYSASM`, `SYSDBA`, and `SYSOPER` privileges as derived from the password file.

Migrating a Database to Use ASM

With a new installation of Oracle Database and ASM, you can initially create your database and select the ASM storage option. If you have an existing Oracle database that stores database files in the operating system file system or on raw devices, then you can migrate some or all of your datafiles to ASM storage.

Oracle provides several methods for migrating your database to ASM. Using ASM will enable you to realize the benefits of automation and simplicity in managing your database storage. You can use the following methods to migrate to ASM as described in this section:

- Using Oracle Enterprise Manager to Migrate Databases to ASM
- Manually Migrating to ASM Using Oracle Recovery Manager
- Migrating to ASM Best Practices White Papers on Oracle Technology Network (OTN)

Note: You must upgrade to at least Oracle Database 10*g* before migrating your database to ASM.

Using Oracle Enterprise Manager to Migrate Databases to ASM

Enterprise Manager enables you to perform cold and hot database migration with a GUI. You can access the migration wizard from the Enterprise Manager Home page under the Change Database heading.

See Also: Chapter 6, "Administering ASM with Oracle Enterprise Manager" for more information about using Enterprise Manager to upgrade to ASM

Manually Migrating to ASM Using Oracle Recovery Manager

You can use Oracle Recovery Manager (RMAN) to manually migrate to ASM. You can also use RMAN to migrate a single tablespace or datafile to ASM.

See Also: *Oracle Database Backup and Recovery User's Guide*. for detailed instructions about migrating ASM data using RMAN

Migrating to ASM Best Practices White Papers on Oracle Technology Network (OTN)

The Oracle Maximum Availability Architecture (MAA) Web site provides excellent best practices technical white papers based on different scenarios, such as:

- Minimal Downtime Migration to ASM
- Platform Migration using Transportable Tablespaces
- Platform Migration using Transportable Database

See Also: For information about Oracle ASM best practices for migrating to Oracle ASM from environments that do not use Oracle ASM, refer to the documentation at the MAA link on Oracle Technology Network:

`http://www.oracle.com/technetwork/database/features/availability/maa-096107.html`

4 Administering ASM Disk Groups

This chapter describes how to administer Automatic Storage Management (ASM) disk groups. The examples in this chapter use SQL statements and assume that an ASM instance is running. This chapter contains the following topics:

- Managing Automatic Storage Management (ASM) Disk Groups
- ASM Disk Discovery
- Managing Capacity in Disk Groups
- ASM Mirroring and Disk Group Redundancy
- Performance and Scalability Considerations for Disk Groups
- Disk Group Compatibility
- Mounting and Dismounting Disk Groups
- Checking the Internal Consistency of Disk Group Metadata
- Dropping Disk Groups
- Using Views to Obtain ASM Information

For information about starting up an ASM instance, refer to "Starting Up an ASM Instance" on page 3-10. For information about administering ASM disk groups with Enterprise Manager, refer to Chapter 6, "Administering ASM with Oracle Enterprise Manager". For information about administering ASM disk groups with ASMCMD, refer to Chapter 7, "ASM Command-Line Utility". For information about administering ASM disk groups with DBCA, refer to your platform-specific Oracle Database installation guide.

> **See Also:** The Oracle Cloud Storage page on the Oracle Technology Network Web site at `http://www.oracle.com/technetwork/database/cloud-storage/index.html` for more information about Oracle ASM

Managing Automatic Storage Management (ASM) Disk Groups

This section explains how to create, alter, drop, mount, and dismount Automatic Storage Management (ASM) disk groups. Note that database instances that use ASM can continue operating while you administer disk groups. This section contains the following topics:

- Creating Disk Groups
- Altering Disk Groups

Creating Disk Groups

The CREATE DISKGROUP SQL statement is used to create disk groups. When creating a disk group, you need to:

- Assign a unique name to the disk group.
- Specify the redundancy level of the disk group.

  If you want ASM to mirror files, you specify the redundancy level as NORMAL REDUNDANCY (2-way mirroring by default for most file types) or HIGH REDUNDANCY (3-way mirroring for all files). You specify EXTERNAL REDUNDANCY if you do not want mirroring by ASM. For example, you might choose EXTERNAL REDUNDANCY if you want to use storage array protection features.

  For more information about redundancy levels, refer to "ASM Mirroring and Failure Groups" on page 4-13.
- Specify the disks as belonging to specific failure groups.

  For information about failure groups, refer to "Understanding ASM Concepts" on page 1-2 and "ASM Mirroring and Failure Groups" on page 4-13.
- Specify the disks that are to be formatted as ASM disks belonging to the disk group.

  The disks can be specified using operating system dependent wildcard characters in search strings that ASM then uses to find the disks. You can specify names for the disks with the NAME clause or use the system-generated names.
- Optionally specify disk group attributes, such software compatibility or allocation unit size.

ASM programmatically determines the size of each disk. If for some reason this is not possible, or if you want to restrict the amount of space used on a disk, you are able to specify a SIZE clause for each disk. ASM creates operating system–independent names for the disks in a disk group that you can use to reference the disks in other SQL statements. Optionally, you can provide your own name for a disk using the NAME clause. Disk names are available in the V$ASM_DISK view.

Note: A disk cannot belong to multiple disk groups.

The ASM instance ensures that any disk in a newly created disk group is addressable and is not currently a member of another disk group. You must use FORCE only when adding a disk that was dropped FORCE. If a disk is dropped NOFORCE, then use can add it NOFORCE. For example, a disk might have failed and was dropped from its disk group. After the disk is repaired, it is no longer part of any disk group, but ASM still recognizes that the disk had been a member of a disk group. You must use the FORCE flag to include the disk in a new disk group. In addition, the disk must be addressable, and the original disk group must not be mounted. Otherwise, the operation fails.

Note: Use caution when using the FORCE option to add a previously used disk to a disk group; you might cause another disk group to become unusable.

The CREATE DISKGROUP statement mounts the disk group for the first time, and adds the disk group name to the ASM_DISKGROUPS initialization parameter if a server parameter file is being used. If a text initialization parameter file is being used and you want the disk group to be automatically mounted at instance startup, then you must remember to add the disk group name to the ASM_DISKGROUPS initialization parameter before the next time that you shut down and restart the ASM instance. You can also create disk groups with Oracle Enterprise Manager. Refer to "Creating Disk Groups" on page 6-5.

See Also:

- The CREATE DISKGROUP SQL statement in the *Oracle Database SQL Language Reference*
- For information about using ASMLib when creating disk groups, refer to the Oracle ASMLib page on the Oracle Technology Network Web site at http://www.oracle.com/technetwork/topics/linux/asmlib/index-101839.html

Example: Creating a Disk Group

The following examples assume that the ASM_DISKSTRING initialization parameter is set to '/devices/*' and ASM disk discovery identifies the following disks in the /devices directory.

Controller 1:

```
/devices/diska1
/devices/diska2
/devices/diska3
/devices/diska4
```

Controller 2:

```
/devices/diskb1
/devices/diskb2
/devices/diskb3
/devices/diskb4
```

The SQL statement in Example 4–1 creates a disk group named dgroup1 with normal redundancy consisting of two failure groups controller1 or controller2 with four disks in each failure group.

Example 4–1 Creating a Disk Group

```
% SQLPLUS /NOLOG
SQL> CONNECT / AS SYSASM
Connected to an idle instance.
SQL> STARTUP NOMOUNT

CREATE DISKGROUP dgroup1 NORMAL REDUNDANCY
FAILGROUP controller1 DISK
'/devices/diska1' NAME diska1,
'/devices/diska2' NAME diska2,
'/devices/diska3' NAME diska3,
'/devices/diska4' NAME diska4
FAILGROUP controller2 DISK
'/devices/diskb1' NAME diskb1,
'/devices/diskb2' NAME diskb2,
'/devices/diskb3' NAME diskb3,
'/devices/diskb4' NAME diskb4
ATTRIBUTE 'au_size'='4M',
          'compatible.asm' = '11.1',
    'compatible.rdbms' = '11.1';
```

In Example 4–1, the NAME clauses enable you to explicitly assign names to the disks rather than the default system-generated names. The system-generated names are in the form *diskgroupname_nnnn*, where *nnnn* is the disk number for the disk in the disk group. For ASMLIB disks, the disk name defaults to the ASMLIB name that is the user label of the disk; for example, mydisk is the default ASM disk name for ORCL:mydisk.

When creating the disk group, the values of following disk group attributes were explicitly set:

- AU_SIZE specifies the size of the allocation unit for the disk group. For information about allocation unit size and extents, see "Extents" on page 1-7.
- COMPATIBLE.ASM determines the minimum software version for any ASM instance that uses a disk group. For information about the COMPATIBLE.ASM attribute, see "COMPATIBLE.ASM" on page 4-22.
- COMPATIBLE.RDBMS determines the minimum software version for any database instance that uses a disk group. For information about the COMPATIBLE.RDBMS attribute, see "COMPATIBLE.RDBMS" on page 4-22.

Altering Disk Groups

You can use the ALTER DISKGROUP statement to alter a disk group configuration. You can add, resize, or drop disks while the database remains online. Whenever possible, multiple operations in a single ALTER DISKGROUP statement are recommended.

ASM automatically rebalances when the configuration of a disk group changes. By default, the ALTER DISKGROUP statement does not wait until the operation is complete before returning. Query the V$ASM_OPERATION view to monitor the status of this operation.

You can use the REBALANCE WAIT clause if you want the ALTER DISKGROUP statement processing to wait until the rebalance operation is complete before returning. This is especially useful in scripts. The statement also accepts a REBALANCE NOWAIT clause that invokes the default behavior of conducting the rebalance operation asynchronously in the background.

You can interrupt a rebalance running in wait mode by typing CTRL-C on most platforms. This causes the statement to return immediately with the message ORA-01013: user requested cancel of current operation, and then to continue the operation asynchronously. Typing CTRL-C does not cancel the rebalance operation or any disk add, drop, or resize operations.

To control the speed and resource consumption of the rebalance operation, you can include the REBALANCE POWER clause in statements that add, drop, or resize disks. Refer to "Manually Rebalancing Disk Groups" on page 4-8 for more information about this clause.

This section contains the following topics:

- Adding Disks to a Disk Group
- Dropping Disks from Disk Groups
- Resizing Disks in Disk Groups
- Undropping Disks in Disk Groups
- Manually Rebalancing Disk Groups
- Tuning Rebalance Operations

See Also: The ALTER DISKGROUP SQL statement in the *Oracle Database SQL Language Reference*

Adding Disks to a Disk Group

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. For an example of the CREATE DISKGROUP SQL statement, refer to Example 4–1 on page 4-3. After you add new disks, the new disks gradually begin to accommodate their share of the workload as rebalancing progresses.

ASM behavior when adding disks to a disk group is best illustrated through"Example: Adding Disks to a Disk Group" on page 4-5. You can also add disks to a disk group with Oracle Enterprise Manager, described in "Adding Disks to Disk Groups" on page 6-7.

Example: Adding Disks to a Disk Group The statements presented in this example demonstrate the interactions of disk discovery with the ADD DISK operation.

Assume that disk discovery identifies the following disks in directory /devices:

/devices/diska1 -- member of dgroup1
/devices/diska2 -- member of dgroup1
/devices/diska3 -- member of dgroup1
/devices/diska4 -- member of dgroup1
/devices/diska5 -- candidate disk
/devices/diska6 -- candidate disk
/devices/diska7 -- candidate disk
/devices/diska8 -- candidate disk

/devices/diskb1 -- member of dgroup1
/devices/diskb2 -- member of dgroup1
/devices/diskb3 -- member of dgroup1
/devices/diskb4 -- member of dgroup2

/devices/diskc1 -- member of dgroup2
/devices/diskc2 -- member of dgroup2
/devices/diskc3 -- member of dgroup3
/devices/diskc4 -- candidate disk

/devices/diskd1 -- candidate disk
/devices/diskd2 -- candidate disk
/devices/diskd3 -- candidate disk
/devices/diskd4 -- candidate disk
/devices/diskd5 -- candidate disk
/devices/diskd6 -- candidate disk
/devices/diskd7 -- candidate disk
/devices/diskd8 -- candidate disk

You can query the V$ASM_DISK view to display the status of ASM disks. See "Using Views to Obtain ASM Information" on page 4-29.

The following statement would fail because /devices/diska1 - /devices/diska4 already belong to dgroup1.

```
ALTER DISKGROUP dgroup1 ADD DISK
     '/devices/diska*';
```

The following statement would successfully add disks /devices/diska5 through /devices/diska8 to dgroup1. Because no FAILGROUP clauses are included in the ALTER DISKGROUP statement, each disk is assigned to its own failure group. The NAME clauses assign names to the disks, otherwise they would have been assigned system-generated names.

```
ALTER DISKGROUP dgroup1 ADD DISK
     '/devices/diska5' NAME diska5,
     '/devices/diska6' NAME diska6,
     '/devices/diska7' NAME diska7,
     '/devices/diska8' NAME diska8,
```

The following statement would fail because the search string matches disks that are contained in other disk groups. Specifically, /devices/diska4 belongs to disk group dgroup1 and /devices/diskb4 belongs to disk group dgroup2.

```
ALTER DISKGROUP dgroup1 ADD DISK
     '/devices/disk*4';
```

The following statement would successfully add /devices/diskd1 through /devices/diskd8 to disk group dgroup1. This statement runs with a rebalance power of 5, and does not return until the rebalance operation is complete.

```
ALTER DISKGROUP dgroup1 ADD DISK
      '/devices/diskd*'
       REBALANCE POWER 5 WAIT;
```

If /devices/diskc3 was previously a member of a disk group that no longer exists, then you could use the FORCE option to add them as members of another disk group. For example, the following use of the FORCE clause enables /devices/diskc3 to be added to dgroup2, even though it is a current member of dgroup3. For this statement to succeed, dgroup3 cannot be mounted.

```
ALTER DISKGROUP dgroup2 ADD DISK
     '/devices/diskc3' FORCE;
```

Dropping Disks from Disk Groups

To drop disks from a disk group, use the DROP DISK clause of the ALTER DISKGROUP statement. You can also drop all of the disks in specified failure groups using the DROP DISKS IN FAILGROUP clause.

When a disk is dropped, the disk group is rebalanced by moving all of the file extents from the dropped disk to other disks in the disk group. A drop disk operation might fail if not enough space is available on the other disks. The best approach is to perform both the add and drop operation with the same ALTER DISKGROUP statement. This has the benefit of rebalancing data extents once and ensuring that there is enough space for the rebalance operation to succeed.

> **Caution:** 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. For more information, refer to the *Oracle Database SQL Language Reference* and the *Oracle Database Reference*.

If you specify the FORCE clause for the drop operation, the disk is dropped even if ASM cannot read or write to the disk. You cannot use the FORCE flag when dropping a disk from an external redundancy disk group.

> **Caution:** A DROP FORCE operation leaves data at reduced redundancy for as long as it takes for the subsequent rebalance operation to complete. This increases your exposure to data loss if there is a subsequent disk failure during rebalancing. DROP FORCE should be used only with great care.

You can also drop disks from a disk group with Oracle Enterprise Manager. See "Dropping Disks from Disk Groups" on page 6-9.

Example: Dropping Disks from Disk Groups The statements in this example demonstrate how to drop disks from the disk group dgroup1 described in "Example: Adding Disks to a Disk Group" on page 4-5.

The following example drops diska5 from disk group dgroup1.

```
ALTER DISKGROUP dgroup1 DROP DISK diska5;
```

The following example drops diska5 from disk group dgroup1, and also illustrates how multiple actions are possible with one ALTER DISKGROUP statement.

```
ALTER DISKGROUP dgroup1 DROP DISK diska5
     ADD FAILGROUP failgrp1 DISK '/devices/diska9' NAME diska9;
```

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 you do not specify a new size in the SIZE clause then ASM uses the size of the disk as returned by the operating system. The new size is written to the 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.

Example: Resizing Disks in Disk Groups The following example resizes all of the disks in failure group failgrp1 of disk group dgroup1. If the new size is greater than disk capacity, the statement will fail.

```
ALTER DISKGROUP dgroup1
     RESIZE DISKS IN FAILGROUP failgrp1 SIZE 100G;
```

Undropping Disks in Disk Groups

The UNDROP DISKS clause of the ALTER DISKGROUP statement enables you to cancel all pending drops of disks within disk groups. If a drop disk operation has already completed, then this statement cannot be used to restore it. 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.

Example: Undropping Disks in Disk Groups The following example cancels the dropping of disks from disk group dgroup1:

```
ALTER DISKGROUP dgroup1 UNDROP DISKS;
```

Manually Rebalancing Disk Groups

You can manually rebalance the files in a disk group using the REBALANCE clause of the ALTER DISKGROUP statement. This would normally not be required, because ASM automatically rebalances disk groups when their configuration changes. You might want to do a manual rebalance operation if you want to control the speed of what would otherwise be an automatic rebalance operation.

The POWER clause of the ALTER DISKGROUP...REBALANCE statement specifies the degree of parallelism, and thus the speed of the rebalance operation. It can be set to a value from 0 to 11. A value of 0 halts a rebalancing operation until the statement is either implicitly or explicitly re-run. The default rebalance power is set by the ASM_POWER_LIMIT initialization parameter. See "Tuning Rebalance Operations" on page 4-9 for more information.

The power level of an ongoing rebalance operation can be changed by entering the rebalance statement with a new level.

The ALTER DISKGROUP...REBALANCE command by default returns immediately so that you can issue other commands while the rebalance operation takes place asynchronously in the background. You can query the V$ASM_OPERATION view for the status of the rebalance operation.

If you want the ALTER DISKGROUP...REBALANCE command to wait until the rebalance operation is complete before returning, you can add the WAIT keyword to the REBALANCE clause. This is especially useful in scripts. The command also accepts a NOWAIT keyword, which invokes the default behavior of conducting the rebalance operation asynchronously. You can interrupt a rebalance running in wait mode by typing CTRL-C on most platforms. This causes the command to return immediately with the message ORA-01013: user requested cancel of current operation, and then to continue the rebalance operation asynchronously.

Additional rules for the rebalance operation include the following:

- An ongoing rebalance command will be restarted if the storage configuration changes either when you alter the configuration, or if the configuration changes due to a failure or an outage. Furthermore, if the new rebalance fails because of a user error, then a manual rebalance may be required.
- The ALTER DISKGROUP...REBALANCE statement runs on a single node even if you are using Oracle Real Application Clusters (Oracle RAC).
- ASM can perform one disk group rebalance at a time on a given instance. Therefore, if you have initiated multiple rebalances on different disk groups, then Oracle processes this operation serially. However, you can initiate rebalances on different disk groups on different nodes in parallel.
- Rebalancing continues across a failure of the ASM instance performing the rebalance.
- The REBALANCE clause (with its associated POWER and WAIT/NOWAIT keywords) can also be used in ALTER DISKGROUP commands that add, drop, or resize disks.

Note: Oracle will restart the processing of an ongoing rebalance operation if the storage configuration changes. Furthermore, if the next rebalance operation fails because of a user error, then you may need to perform a manual rebalance.

Example: Manually Rebalancing a Disk Group The following example manually rebalances the disk group dgroup2. The command does not return until the rebalance operation is complete.

```
ALTER DISKGROUP dgroup2 REBALANCE POWER 5 WAIT;
```

For more information about rebalancing operations, refer to "Tuning Rebalance Operations" on page 4-9.

Tuning Rebalance Operations

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 higher the power limit, the more quickly a rebalance operation can complete. 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.

The default value of 1 minimizes disruption to other applications. The appropriate value is dependent on your hardware configuration, performance requirements, and availability requirements

If a rebalance is in progress because a disk is manually or automatically 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.

The V$ASM_OPERATION view provides information for adjusting ASM_POWER_LIMIT and the resulting power of rebalance operations. The V$ASM_OPERATION view also gives an estimate in the EST_MINUTES column of the amount of time remaining for the rebalance operation to complete. You can see the effect of changing the rebalance power by observing the change in the time estimate.

See Also: "Manually Rebalancing Disk Groups" on page 4-8 for more information.

ASM Disk Discovery

Disk discovery is the mechanism used to find the operating system names for disks ASM can access. It is used to find all the disks that comprise a disk group to be mounted, the disks an administrator wants to add to a disk group, or the disks the administrator might consider adding to a disk group. This section contains the following topics:

- How A Disk is Discovered
- Disk Discovery Rules
- Improving Disk Discovery Time

For additional information about disk discovery and the ASM_DISKSTRING initialization parameter, refer to "ASM_DISKSTRING" on page 3-5.

How A Disk is Discovered

While an ASM instance is initialized, ASM discovers and examines the contents of all of the disks that are in the paths that you designated with values in the ASM_DISKSTRING initialization parameter. Disk discovery also occurs when you:

- Run the ALTER DISKGROUP...ADD DISK and ALTER DISKGROUP...RESIZE DISK commands
- Query the V$ASM_DISKGROUP and V$ASM_DISK views

After ASM successfully discovers a disk, the disk appears in the V$ASM_DISK view. Disks that belong to a disk group, that is, disks that have a disk group name in the disk header, show a status of MEMBER. Disks that were discovered, but that have not yet been assigned to a disk group, have a status of either CANDIDATE or PROVISIONED.

The PROVISIONED status implies that an additional platform-specific action has been taken by an administrator to make the disk available for ASM. For example, on Windows computers, the administrator might have used asmtool or asmtoolg to stamp the disk with a header. On Linux computers, the administrator might have used ASMLIB to prepare the disk for ASM.

The following SQL query shows one candidate and six member disks:

```
SELECT name, header_status, path FROM V$ASM_DISK;

NAME      HEADER_STATUS PATH
--------- ------------- --------------------
          CANDIDATE     /dev/rdsk/disk07
DISK06    MEMBER        /dev/rdsk/disk06
DISK05    MEMBER        /dev/rdsk/disk05
DISK04    MEMBER        /dev/rdsk/disk04
DISK03    MEMBER        /dev/rdsk/disk03
DISK02    MEMBER        /dev/rdsk/disk02
DISK01    MEMBER        /dev/rdsk/disk01

7 rows selected.
```

Disk Discovery Rules

The rules for discovering ASM disks are as follows:

- ASM can discover up to 10,000 disks. That is, if more than 10,000 disks match the ASM_DISKSTRING initialization parameter, then ASM discovers only the first 10,000.
- ASM only discovers disks that contain a partition table.

> **Note:** ASM does not discover a disk that contains an operating system partition table, even if the disk is in an ASM disk string search path and ASM has read and write permission for the disk.

- When adding a disk, the FORCE option must be used if ASM recognizes that the disk was managed by Oracle. Such a disk appears in the V$ASM_DISK view with a status of FOREIGN. In this case, you can only add the disk to a disk group by using the FORCE keyword.

In addition, ASM identifies the following configuration errors during discovery:

- Multiple paths to the same disk

In this case, if the disk is part of a disk group, then disk group mount fails. If the disk is being added to a disk group with the ADD DISK or CREATE DISKGROUP command, then the command fails. To correct the error, adjust the ASM_DISKSTRING value so that ASM will not discover multiple paths to the same disk. Or if you are using multipathing software, then ensure that you include only the pseudo-device name in the ASM_DISKSTRING value. See "ASM and Multipathing" on page 2-2.

- Multiple ASM disks with the same disk header

  This can be caused by having copied one disk onto another. In this case, the disk group mount operation fails.

Improving Disk Discovery Time

The value for the ASM_DISKSTRING initialization parameter is an operating system–dependent value that ASM uses to limit the set of paths that the discovery process uses to search for disks. When a new disk is added to a disk group, each ASM instance that has the disk group mounted must be able to discover the new disk using its ASM_DISKSTRING.

In many cases, the default value (NULL) is sufficient. Using a more restrictive value might reduce the time required for ASM to perform discovery, and thus improve disk group mount time or the time for adding a disk to a disk group. Oracle might need to dynamically change the ASM_DISKSTRING before adding a disk so that the new disk will be discovered through this parameter.

Note that the default value of ASM_DISKSTRING might not find all disks in all situations. If your site is using a third-party vendor ASMLIB, then the vendor might have discovery string conventions that you must use for ASM_DISKSTRING. In addition, if your installation uses multipathing software, then the software might place pseudo-devices in a path that is different from the operating system default. See "ASM and Multipathing" on page 2-2 and consult the multipathing vendor documentation for details.

Managing Capacity in Disk Groups

When ASM provides redundancy, such as when you create a disk group with NORMAL or HIGH redundancy, you must have sufficient 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 not enough space remains, then some files might end up with reduced redundancy.

Reduced redundancy means that 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 file extent with two or fewer total copies of the extent instead of three. In the case of unprotected files, data extents could be missing altogether. 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.

The following guidelines help ensure that you have sufficient space to restore full redundancy for all disk group data after the failure of one or more disks.

- In a 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.

- In a 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.

> **Note:** When you lose multiple disks from multiple failure groups, then you could lose both the primary and the redundant copies of your data. In addition, if you do not have enough capacity to restore redundancy, then ASM can continue to operate. However, if another disk fails, then the system may not be able to tolerate additional failures.

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

  For a normal redundancy disk group, the value is the total raw space for all of the disks in the largest failure group. The largest failure group is the one with the largest total raw capacity. For example, if each disk is in its own failure group, then the value would be the size of the largest capacity disk.

  For a high redundancy disk group, the value is the total raw space for all of the disks in the two largest failure groups.

- `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 in size.

- `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 ASM 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. Therefore, there may be situations where the value in the `FREE_MB` column shows unused capacity but because one of the ASM disks is full, database writes fail because of the imbalance in the disk group. Make sure 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. Therefore, the datafile 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.

> **Note:** The values in the TOTAL_MB and FREE_MB columns best describe space usage when you do not configure ASM mirroring, that is, when you use external redundancy.

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:

```
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
------------ ------ ---------- ---------- ----------------------- --------------
DISKGROUP1   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. Note that 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. Note in the example that:

```
FREE_MB - REQUIRED_MIRROR_FREE_MB = 2 * USABLE_FILE_MB
```

```
3768 - 1024 = 2 * 1372 = 2744
```

Negative Values of USABLE_FILE_MB

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

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

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

ASM Mirroring and Disk Group Redundancy

This section contains the following topics:

- ASM Mirroring and Failure Groups
- ASM Recovery from Read and Write I/O Errors
- ASM Fast Mirror Resync
- Preferred Read Failure Groups

ASM Mirroring and Failure Groups

If you specify mirroring for a file, then ASM automatically stores redundant copies of the file extents in separate failure groups. Failure groups apply only to normal and high redundancy disk groups. You can define the failure groups for each disk group when you create or alter the disk group.

There are three types of disk groups based on the ASM redundancy level. Table 4–1 lists the types with their supported and default mirroring levels. The default mirroring levels indicate the mirroring level with which each file is created unless a different mirroring level is designated.

Table 4–1 Mirroring Options for ASM Disk Group Types

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

The redundancy level controls how many disk failures are tolerated without dismounting the disk group or losing data. Each file is allocated based on its own redundancy, but the default comes from the disk group. The redundancy levels are:

- External redundancy

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

- Normal redundancy

  ASM provides two-way mirroring. By default all files are mirrored so that there are two copies of every **data extent**. A loss of one ASM disk is tolerated.

- High redundancy

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

Failure groups enable the mirroring of metadata and user data. System reliability can diminish if your environment has an insufficient number of failure groups.

This section contains these topics:

- ASM Failure Groups
- How ASM Manages Disk Failures
- Guidelines for Using Failure Groups
- Failure Group Frequently Asked Questions

ASM Failure Groups

Failure groups are used to store mirror copies of data. When ASM allocates an extent for a normal redundancy file, ASM allocates a primary copy and a secondary copy. ASM chooses the disk on which to store the secondary copy so that it is in a different failure group than the primary copy. Each copy is on a disk in a different failure group so that the simultaneous failure of all disks in a failure group does not result in data loss.

A failure group is a subset of the disks in a disk group, which could fail at the same time because they share hardware. The failure of common hardware must be tolerated. Four drives that are in a single removable tray of a large JBOD array should be in the same failure group because the tray could be removed 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. However, ASM mirroring is not intended to protect against a fire in the computer room that destroys the entire cabinet.

There are always failure groups even if they are not explicitly created. If you do not specify a failure group for a disk, then Oracle automatically creates a new failure group containing just that disk. A normal redundancy disk group must contain at least two failure groups. A high redundancy disk group must contain at least three failure groups. However, Oracle recommends using several failure groups. A small number of failure groups, or failure groups of uneven capacity, can create allocation problems that prevent full use of all of the available storage.

How ASM Manages Disk Failures

Depending on the redundancy level of a disk group and how you define failure groups, the failure of one or more disks could result in either of the following:

- The disks are first taken offline and then automatically dropped. In this case, the disk group remains mounted and serviceable. In addition, because of mirroring, all of the disk group data remains accessible. After the disk drop operation, ASM performs a rebalance to restore full redundancy for the data on the failed disks.
- The entire disk group is automatically dismounted, which means loss of data accessibility.

Guidelines for Using Failure Groups

The following are guidelines for using failure groups:

- Each disk in a disk group can belong to only one failure group.
- Failure groups should all be of the same size. Failure groups of different sizes may lead to reduced availability.
- ASM requires at least two failure groups to create a normal redundancy disk group and at least three failure groups to create a high redundancy disk group.

Failure Group Frequently Asked Questions

This section discusses frequently asked questions about failure group under the following topics:

- How Many Failure Groups Should I Create?
- How are Multiple Failure Groups Recovered after Simultaneous Failures?
- When Should External, Normal, or High Redundancy Be Used?

How Many Failure Groups Should I Create?

Choosing the number of failure groups to create depends on the types of failures that need to be tolerated without data loss. For small numbers of disks, such as fewer than 20, it is usually best to use the default failure group creation that puts every disk in its own failure group.

Using the default failure group creation for small numbers of disks is also applicable for large numbers of disks where your main concern is disk failure. For example, a disk group might be configured from several small modular disk arrays. If the system must continue operating when an entire modular array fails, then a failure group should consist of all of the disks in one module. If one module fails, then all of the data on that module will be relocated to other modules to restore redundancy. Disks should be placed in the same failure group if they depend on a common piece of hardware whose failure must be tolerated with no loss of availability.

How are Multiple Failure Groups Recovered after Simultaneous Failures?

A simultaneous failure can occur if there is a failure of a piece of hardware used by multiple failure groups. This type of failure usually forces a dismount of the disk group if all disks are unavailable.

When Should External, Normal, or High Redundancy Be Used?

ASM mirroring runs on the database server and Oracle recommends to off load this processing to the storage hardware RAID controller by using external redundancy. You can use normal redundancy in the following scenarios:

- Storage system does not have RAID controller
- Mirroring across storage arrays
- Extended cluster configurations

In general, ASM mirroring is the Oracle alternative to third party logical volume managers. ASM mirroring eliminates the need to deploy additional layers of software complexity in your Oracle database environment.

ASM Recovery from Read and Write I/O Errors

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

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

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

When encountering a write error, a database instance sends the ASM instance a *disk offline* message.

- If database can successfully complete a write to at least one extent copy and receive acknowledgment of the offline disk from ASM, the write is considered successful.
- If the write to all mirror side fails, database takes the appropriate actions in response to a write error such as taking the tablespace offline.

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

The ASMCMD `remap` command was introduced to address situations where a range of bad sectors exists on a disk and must be corrected before ASM or database I/O. For information on the `remap` command, see "remap Command" on page 7-20.

ASM Fast Mirror Resync

Restoring the redundancy of an ASM disk group after a transient disk path failure can be time consuming. This is especially true if the recovery process requires rebuilding an entire ASM failure group. ASM fast mirror resync significantly reduces the time to resynchronize a failed disk in such situations. When you replace the failed disk, ASM can quickly resynchronize the ASM disk extents.

> **Note:** To use this feature, the disk group compatibility attributes must be set to `11.1` or higher. For more information, refer to "Disk Group Compatibility" on page 4-21.

Any problems that make a failure group temporarily unavailable are considered transient failures that can be recovered by the ASM fast mirror resync feature. For example, transient failures can be caused by disk path malfunctions, such as cable failures, host bus adapter failures, controller failures, or disk power supply interruptions.

ASM fast resync keeps track of pending changes to extents on an `OFFLINE` disk during an outage. The extents are resynced when the disk is brought back online.

By default, ASM drops a disk in 3.6 hours after it is taken offline. You can set the `DISK_REPAIR_TIME` attribute to delay the drop operation by specifying a time interval to repair the disk and bring it back online. The time can be specified in units of minutes (`m` or `M`) or hours (`h` or `H`). If you omit the unit, then the default unit is hours.

If the attribute is not set explicitly, then the default value (`3.6h`) applies to disks that have been set to `OFFLINE` mode without an explicit `DROP AFTER` clause. Note that disks taken offline due to I/O errors do not have a `DROP AFTER` clause.

The default `DISK_REPAIR_TIME` attribute value is an estimate that should be adequate for most environments. However, make sure that the attribute value is set to the amount of time that you think is necessary in your environment to fix any transient disk error and that you are willing to tolerate reduced data redundancy.

The elapsed time (since the disk was set to `OFFLINE` mode) is incremented only when the disk group containing the offline disks is mounted. The `REPAIR_TIME` column of `V$ASM_DISK` shows the amount of time left (in seconds) before an offline disk is dropped. After the specified time has elapsed, ASM drops the disk. You can override this attribute with an `ALTER DISKGROUP DISK OFFLINE` statement and the `DROP AFTER` clause.

> **Note:** If a disk is offlined by ASM because of an I/O (write) error or is explicitly offlined using the `ALTER DISKGROUP... OFFLINE` statement without the `DROP AFTER` clause, then the value that is specified for the `DISK_REPAIR_TIME` attribute for the disk group is used. If this attribute value is changed with the `ALTER DISKGROUP... SET ATTRIBUTE 'disk_repair_time'` statement before this offlined disk is dropped, then the new (current) value of the attribute is used by the ASM disk offline functionality but the elapsed time is not reset. You can confirm this behavior by viewing the ASM alert log.

If an offline disk is taken offline for a second time, then the elapsed time is reset and restarted. If another time is specified with the DROP AFTER clause for this disk, the first value is overridden and the new value applies. A disk that is in OFFLINE mode cannot be dropped with an ALTER DISKGROUP DROP DISK statement; an error is returned if attempted. If for some reason the disk must be dropped (such as the disk cannot be repaired) before the repair time has expired, a disk can be dropped immediately by issuing a second OFFLINE statement with a DROP AFTER clause specifying 0h or 0m.

You can use ALTER DISKGROUP to set the DISK_REPAIR_TIME attribute to a specified hour or minute value, such as 4.5 hours or 270 minutes. For example:

```
ALTER DISKGROUP dg01 SET ATTRIBUTE 'disk_repair_time' = '4.5h'
ALTER DISKGROUP dg01 SET ATTRIBUTE 'disk_repair_time' = '270m'
```

After you repair the disk, run the SQL statement ALTER DISKGROUP DISK ONLINE. This statement brings a repaired disk group back online to enable writes so that no new writes are missed. This statement also starts a procedure to copy of all of the extents that are marked as stale on their redundant copies.

If a disk goes offline when the ASM instance is in rolling upgrade mode, the disk remains offline until the rolling upgrade has ended and the timer for dropping the disk is stopped until the ASM cluster is out of rolling upgrade mode. See "Using ASM Rolling Upgrades" on page 3-15. Examples of taking disks offline and bringing them online follow.

The following example takes disk D3_0001 offline and drops it after five minutes.

```
ALTER DISKGROUP D3 OFFLINE DISK D3_0001 DROP AFTER 5m;
```

The next example takes the disk D3_0001 offline and drops it after the time period designated by DISK_REPAIR_TIME elapses:

```
ALTER DISKGROUP D3 OFFLINE DISK D3_0001;
```

This example takes all of the disk in failure group F2 offline and drops them after the time period designated by DISK_REPAIR_TIME elapses. IF you used a DROP AFTER clause, then the disks would be dropped after the specified time:

```
ALTER DISKGROUP D3 OFFLINE DISKS IN FAILGROUP F2;
```

The next example brings all of the disks in failure group F2 online:

```
ALTER DISKGROUP D3 ONLINE DISKS IN FAILGROUP F2;
```

This example brings only disk D3_0001 online:

```
ALTER DISKGROUP D3 ONLINE DISK D3_0001;
```

This example brings all of the disks in disk group D3 online:

```
ALTER DISKGROUP D3 ONLINE ALL;
```

Querying the V$ASM_OPERATION view while you are running any of these types of ALTER DISKGROUP ... ONLINE statements displays the name and state of the current operation that you are performing. For example, the query:

```
SELECT GROUP_NUMBER, OPERATION, STATE FROM V$ASM_OPERATION;
```

Displays output similar to the following:

```
GROUP_NUMBER OPERA STAT
------------ ----- ----
           1 ONLIN RUN
```

Note that an OFFLINE operation is not displayed in a V$ASM_OPERATION view query.

See Also: *Oracle Database SQL Language Reference* for information about ALTER DISKGROUP, CREATE DISKGROUP, and ASM disk group attributes

Preferred Read Failure Groups

When you configure ASM failure groups, it might be more efficient for a node to read from an extent that is closest to the node, even if that extent is a secondary extent. In other words, you can configure ASM to read from a secondary extent if that extent is closer to the node instead of ASM reading from the primary copy which might be farther from the node. Using preferred read failure groups is most useful in extended clusters.

To use this feature, Oracle recommends that you configure at least one mirrored extent copy from a disk that is local to a node in an extended cluster. However, a failure group that is preferred for one instance might be remote to another instance in the same Oracle RAC database. The parameter setting for preferred read failure groups is instance specific.

Note: By default, when you create a disk group, every disk in the disk group belongs to one failure group. Oracle does not recommend that you configure more than one preferred read failure group for each instance in a disk group. If you configure more than one preferred read failure group for each instance, then Oracle writes messages to an alert log.

Note: To use this feature, the disk group compatibility attributes must be set to 11.1 or higher. For more information, refer to "Disk Group Compatibility" on page 4-21.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for information about configuring preferred read disks in extended clusters

Configuring and Administering Preferred Read Failure Groups

To configure this feature, set the ASM_PREFERRED_READ_FAILURE_GROUPS initialization parameter to specify a list of failure group names as preferred read disks. For more information about this initialization parameter, refer to "ASM_PREFERRED_READ_FAILURE_GROUPS" on page 3-6.

Set the parameter where *diskgroup_name* is the name of the disk group and *failure_group_name* is the name of the failure group, separating these variables with a period. ASM ignores the name of a failure group that you use in this parameter setting if the failure group does not exist in the named disk group. You can append multiple values using commas as a separator as follows:

ASM_PREFERRED_READ_FAILURE_GROUPS = *diskgroup_name.failure_group_name,...*

In an extended cluster, the failure groups that you specify with settings for the ASM_PREFERRED_READ_FAILURE_GROUPS parameter should only contain disks that are local to the instance. For normal redundancy disk groups, there should be only one failure group on each site of the extended cluster.

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

Having more than one failure group on one site can cause the loss of access to the disk group by the other sites if the site containing more than one failure group fails. In addition, by having more than one failure group on a site, an extent might not be mirrored to another site. This can diminish the read performance of the failure group on the other site.

For example, for a normal redundancy disk group, if a site contains two failure groups of a disk group, then ASM might put both mirror copies of an extent on the same site. In this configuration, ASM cannot protect against data loss from a site failure.

You should configure at most two failure groups on a site for a high redundancy disk group. If there are three sites in an extended cluster, for the same reason previously mentioned, then you should only create one failure group.

For a two-site extended cluster, a normal redundancy disk group only has two failure groups. In this case, you can only specify one failure group as a preferred read failure group for each instance.

You can use views to identify preferred read failure groups, such as the V$ASM_DISK view that shows whether a disk is a preferred read disk by the value in the PREFERRED_READ column. You can also use V$ASM_DISK to verify whether local disks in an extended cluster are preferred read disks. Use the ASM disk I/O statistics to verify that read operations are using the preferred read disks that you configured.

If a disk group is not optimally configured for an extended cluster, then ASM records warning messages in the alert logs. To identify specific performance issues with ASM preferred read failure groups, use the V$ASM_DISK_IOSTAT view. This view displays disk I/O statistics for each ASM client. You can also query the V$ASM_DISK_IOSTAT view on a database instance. However, this query only shows the I/O statistics for the database instance. In general, optimal preferred read extended cluster configurations balance performance with disk group availability.

> **See Also:** *Oracle Database Reference* for details about the V$ASM* dynamic performance views

Performance and Scalability Considerations for Disk Groups

This section discusses the following considerations for evaluating disk group performance:

- Determining the Number of Disk Groups
- Performance Characteristics When Grouping Disks
- ASM Storage Limits

Determining the Number of Disk Groups

Use the following criteria to determine the number of disk groups to create:

- Disks in a given disk group should have similar size and performance characteristics. If you have several different types of disks in terms of size and performance, then create several disk groups that contain similar characteristics.

- Create separate disk groups for your database files and flash recovery area for backup files. This configuration allows fast recovery in case of a disk group failure.

Performance Characteristics When Grouping Disks

ASM load balances the file activity by uniformly distributing file extents across all of the disks in a disk group. For this technique to be effective it is important that disks in a disk group be of 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 flash recovery area.

There might be situations where it is acceptable to temporarily have disks of different sizes and performance co-existing in a disk group. This would be the case when migrating from an old set of disks to a new set of disks. The new disks would be added and the old disks dropped. As the old disks are dropped, their storage is migrated to the new disks while the disk group is online.

ASM Storage Limits

ASM has the following limits:

- 63 disk groups in a storage system
- 10,000 ASM disks in a storage system
- 4 PB maximum storage for each ASM disk
- 40 exabyte maximum storage for each storage system
- 1 million files for each disk group

Oracle Database supports datafile 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 are as follows:

- External redundancy - 140 PB
- Normal redundancy - 42 PB
- High redundancy - 15 PB

Disk Group Compatibility

This section describes disk group compatibility under the following topics:

- Overview of Disk Group Compatibility
- Disk Group Compatibility Attributes
- Setting Disk Group Compatibility Attributes
- Considerations When Setting Disk Group Compatibility
- Considerations When Setting Disk Group Compatibility in Replicated Environments

Overview of Disk Group Compatibility

The disk group compatibility feature enables environments to interoperate when they use disk groups from both Oracle Database 10*g* and Oracle Database 11*g*. For example, disk group compatibility settings that are set to Oracle Database 10*g* enable a Database 10*g* database client to access a disk group created with Oracle Database 11*g* ASM.

The disk group attributes that determine compatibility are COMPATIBLE.ASM and COMPATIBLE.RDBMS. The COMPATIBLE.ASM and COMPATIBLE.RDBMS attribute settings determine the minimum Oracle Database software version numbers that a system can use for ASM and RDBMS instance types respectively. For example, if the ASM compatibility setting is 11.1, and RDBMS compatibility is set to 10.1, then the ASM software version must be at least 11.1, and the Oracle Database client software version must be at least 10.1.

When you create a disk group, you can specify the COMPATIBLE.ASM and COMPATIBLE.RDBMS attribute settings in the CREATE DISKGROUP SQL statement. The ALTER DISKGROUP SQL statement can update the compatible attribute settings for existing disk groups. If not specified, 10.1 is the default setting for both the COMPATIBLE.ASM and COMPATIBLE.RDBMS attributes for ASM in Oracle Database 11*g*. See Table 4–2 on page 4-23 for valid combinations of compatible attribute settings.

Advancing the disk group RDBMS and ASM compatibility settings enables you to use the new ASM features that are available in a later release. For example, a disk group with the RDBMS and ASM compatibility set to 11.1 can take advantage of new Oracle 11*g* features, such as variable extent sizes and fast mirror resync. See Table 4–3 on page 4-26 for the features enabled for combinations of compatibility attribute settings.

Note:

- The compatible attribute settings for a disk group can only be advanced; you cannot revert to a lower compatibility setting.
- The disk group compatibility settings determine whether your environment can use the latest ASM features.

Disk Group Compatibility Attributes

The COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes specify the compatibility settings for Oracle ASM and database instances. These attributes are described under the following topics:

- COMPATIBLE.ASM
- COMPATIBLE.RDBMS

COMPATIBLE.ASM

The value for the disk group COMPATIBLE.ASM attribute determines the minimum software version for an ASM instance that uses the disk group. This setting also determines the format of the data structures for the ASM metadata on the disk. The format of the file contents is determined by the database instance. For ASM in Oracle Database 11*g*, 10.1 is the default setting for the COMPATIBLE.ASM attribute.

COMPATIBLE.RDBMS

The value for the disk group COMPATIBLE.RDBMS attribute determines the minimum COMPATIBLE database initialization parameter setting for any database instance that uses the disk group. For example, if the database COMPATIBLE initialization parameter is set to 11.1.0, then COMPATIBLE.RDBMS can be set to any value between 10.1 and 11.1 inclusively. For ASM in Oracle Database 11*g*, 10.1 is the default setting for the COMPATIBLE.RDBMS attribute.

Caution: If you advance the COMPATIBLE.RDBMS attribute, then you *cannot* revert to the previous setting. Before advancing the COMPATIBLE.RDBMS attribute, ensure that the values for the COMPATIBLE initialization parameter for all of the databases that use the disk group are set to at least the new setting for COMPATIBLE.RDBMS before you advance the attribute value.

Note: The database initialization parameter COMPATIBLE enables you to use a new release of Oracle, while at the same time guaranteeing backward compatibility with an earlier release. See *Oracle Database Reference* for more information about the COMPATIBLE initialization parameter.

Setting Disk Group Compatibility Attributes

You can set disk group compatibility with the CREATE DISKGROUP or ALTER DISKGROUP SQL statement.

This section contains these topics:

- Valid Combinations of Compatibility Attribute Settings
- Using CREATE DISKGROUP with Compatibility Attributes
- Using ALTER DISKGROUP with Compatibility Attributes
- Viewing Compatibility Attribute Settings

When setting the values for the COMPATIBLE.RDBMS and COMPATIBLE.ASM attributes, specify at least the first two digits of a valid Oracle Database release number. For example, you can specify compatibility as '10.1' or '11.1'; Oracle assumes that any missing version number digits are zeros.

Note: 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.

See Also: *Oracle Database SQL Language Reference* for more information about the disk group compatibility SQL statements

Valid Combinations of Compatibility Attribute Settings

Table 4–2 shows the valid combinations of the COMPATIBLE.ASM and the COMPATIBLE.RDBMS attributes and the valid ASM and database instance versions for each combination.

Table 4–2 Disk Group Compatibility Attribute Settings Matrix

COMPATIBLE.ASM	COMPATIBLE.RDBMS	ASM Instance Version	COMPATIBLE Setting for RDBMS Instance
10.1	10.1	>=10.1	>=10.1
11.1	10.1	>=11.1	>=10.1
11.1	10.2	>=11.1	>=10.2

Table 4–2 (Cont.) Disk Group Compatibility Attribute Settings Matrix

COMPATIBLE.ASM	COMPATIBLE.RDBMS	ASM Instance Version	COMPATIBLE Setting for RDBMS Instance
11.1	11.1	>=11.1	>=11.0

Based on Table 4–2, these are some of the possible combinations of ASM and database releases:

- The database release is 11.1 (COMPATIBLE set to 11.0) and the ASM release is 10*g*. The ASM functionality remains at 10*g*.
- The database release is 10*g* and the ASM release is 11.1. Both the COMPATIBLE.ASM and the COMPATIBLE.RDBMS disk group attributes are set to the default value of 10.1. The ASM disk group functionality remains at 10*g*.
- The database release is 10*g* and the ASM release is 11.1. COMPATIBLE.ASM is set to 11.1 and COMPATIBLE.RDBMS is set to 10.1. The ASM disk group attributes are displayed in the V$ASM_ATTRIBUTE view. The disk group functionality is described in Table 4–3 on page 4-26.
- The database release is 11.1 (COMPATIBLE set to 11.0) and the ASM release is 11.1. Both the COMPATIBLE.ASM and the COMPATIBLE.RDBMS disk group attributes are set to 11.1. The ASM disk group attributes are displayed in the V$ASM_ATTRIBUTE view. The disk group functionality is described in Table 4–3 on page 4-26.

Note: The V$ASM_ATTRIBUTE does not display any rows when running ASM release 10*g* or when the COMPATIBLE.ASM value is 10.1. Instead you can determine the values for the COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes with the COMPATIBILITY and DATABASE_COMPATIBILITY columns of the V$ASM_DISKGROUP view. The ALLOCATION_UNIT_SIZE column of the V$ASM_DISKGROUP view displays the value of the AU_SIZE disk group attribute.

Using CREATE DISKGROUP with Compatibility Attributes

You can specify the compatibility settings for a disk group with the CREATE DISKGROUP statement when creating the disk group.

The following example creates a normal redundancy disk group diskgrp1 with the ASM compatibility set to 11.1 and the RDBMS compatibility set to the default (assuming that the COMPATIBLE.RDBMS default is less than or equal to 11.1):

```
CREATE DISKGROUP diskgrp1 DISK '/dev/raw/*'
       ATTRIBUTE 'compatible.asm' = '11.1';
```

The following example creates a normal redundancy disk group diskgrp2 with both the ASM and the RDBMS compatibility set to 11.1:

```
CREATE DISKGROUP diskgrp2 DISK '/dev/raw/*'
       ATTRIBUTE 'compatible.rdbms' = '11.1', 'compatible.asm' = '11.1';
```

Using ALTER DISKGROUP with Compatibility Attributes

After a disk group has been created, you can use the ALTER DISKGROUP SQL statement to change the compatibility attributes. Using the ALTER DISKGROUP SQL statement ensures that Oracle can advance the compatibility of the specified disk group before committing the change.

All of the affected databases and file systems should be online when running ALTER DISKGROUP to ensure that advancing compatibility does not reduce the database and file system functionality. When advancing disk group compatibility, advance the COMPATIBLE.ASM attribute before the COMPATIBLE.RDBMS attribute to ensure a valid combination of compatible attribute settings as shown in Table 4–2.

The following example advances the ASM compatibility for disk group diskgrp3 to 11.1. An ASM instance must be at release 11.1 and higher to access the diskgrp3 disk group.

```
ALTER DISKGROUP diskgrp3 SET ATTRIBUTE 'compatible.asm' = '11.1'
```

The following example advances the RDBMS compatibility of the disk group diskgrp4 to 11.1. This example assumes that the ASM compatibility is already advanced to 11.1.

```
ALTER DISKGROUP diskgrp4 SET ATTRIBUTE 'compatible.rdbms' = '11.1'
```

Viewing Compatibility Attribute Settings

In addition to appearing in the V$ASM_ATTRIBUTE view, the compatibility attribute values also appear in the columns labeled DATABASE_COMPATIBILITY and COMPATIBILITY in the V$ASM_DISKGROUP view. See Example 4–2, "Viewing Disk Groups with Associated Attributes" on page 4-30 for an example querying the V$ASM_ATTRIBUTE view.

See Also:

- *Oracle Database SQL Language Reference* for information about disk group attributes
- *Oracle Database Reference* for details on the V$ASM_DISKGROUP and V$ASM_ATTRIBUTE dynamic performance views

Considerations When Setting Disk Group Compatibility

When changing the disk group compatibility settings, there are some considerations that you should be aware of.

- If a backup of a disk group was made with the ASMCMD md_backup command prior to changing the compatibility settings, that full backup file would be incorrect for the updated disk group. Restoring that the full previous backup would set the disk group to the previous compatibility settings.

  You can still use the previous backup to restore some metadata. For example, you could create a new disk group and use the backup file to restore templates and alias directories metadata. For information about using md_backup and md_restore, refer to "md_backup Command" on page 7-17 and "md_restore Command" on page 7-17.
- The disk group compatibility settings should be the same for all replication environments.
- The disk group compatibility settings cannot be changed during a rolling upgrade.
- The compatibility settings can only be advanced and the settings are irreversible.
- Not all combinations of the disk group compatibility settings are valid and some features are not enabled in various combinations. See Table 4–2 on page 4-23 for valid combinations of COMPATIBLE.ASM and COMPATIBLE.RDBMS settings. See Table 4–3 for the disk group features enabled by valid combinations of COMPATIBLE.ASM and COMPATIBLE.RDBMS settings.

Table 4–3 Features Enabled by Disk Group Compatibility Attribute Settings

Disk Group Features Enabled	COMPATIBLE.ASM	COMPATIBLE.RDBMS
Different AU sizes (1, 2, 4, or 8MB)	>= 10.1	=< COMPATIBLE.ASM setting
Attributes are displayed in the `V$ASM_ATTRIBUTE` view	11.1	=< COMPATIBLE.ASM setting
Support for larger AU sizes (16, 32, or 64MB)	11.1	11.1
Fast mirror resync	11.1	11.1
Variable size extents	11.1	11.1
Preferred read failure groups	11.1	11.1

ASM features not explicitly listed in Table 4–3 do not require advancing the disk group compatibility attribute settings.

Considerations When Setting Disk Group Compatibility in Replicated Environments

If you advance disk group compatibility, then you could enable the creation of files that are too large to be managed by a previous Oracle database release. You need to be aware of the file size limits because replicated sites cannot continue using the software from a previous release to manage these large files.

Table 4–4 shows the maximum ASM file sizes supported for `COMPATIBLE.RDBMS` settings. Note that memory consumption is identical in all cases: 280 MB. For Oracle Database 11*g*, the number of extent pointers for each size is 16800, with the largest size using the remainder. The RDBMS instance limits file sizes to a maximum of 128TB.

Table 4–4 Maximum ASM File Size

Redundancy	COMPATIBLE.RDBMS = 10.1	COMPATIBLE.RDBMS = 11.1
External	35 TB	140 PB
Normal	5.8 TB	23 PB
High	3.9 TB	15 PB

Table 4–4 shows that Oracle Database 10*g* can only support a file size of up to 35 TB for external redundancy. If you advance the RDBMS compatibility to `11.1`, then a file can grow beyond 35 TB. However, the larger size will make the file unusable in a replicated and disaster recovery site if the disaster recovery site has a disk group `COMPATIBLE.RDBMS` setting that is incompatible with the larger size.

See Also:

- *Oracle Database Upgrade Guide* for information about database compatibility
- *Oracle Database Administrator's Guide* for information about the `COMPATIBLE` initialization parameter and irreversible compatibility
- *Oracle Database Reference* for information about the `COMPATIBLE` initialization parameter

Mounting and Dismounting Disk Groups

Disk groups that are specified in the ASM_DISKGROUPS initialization parameter are mounted automatically at ASM instance startup. This makes them available to all database instances running on the same node as ASM. The disk groups are dismounted at ASM instance shutdown. ASM also automatically mounts a disk group when you initially create it, and dismounts a disk group if you drop it.

There might be times that you want to mount or dismount disk groups manually. For these actions use the ALTER DISKGROUP...MOUNT or ALTER DISKGROUP...DISMOUNT statement. You can mount or dismount disk groups by name, or specify ALL.

If you try to dismount a disk group that contains open files, the statement will fail, unless you also specify the FORCE clause.

In a clustered ASM environment in RESTRICTED mode, a disk group is mounted in single-instance exclusive mode. No other ASM instance in that cluster can mount that disk group. In this mode the disk group is not usable by any ASM client. Use this mode to perform a fast rebalance.

The following SQL statement dismounts all disk groups that are currently mounted to the ASM instance:

```
ALTER DISKGROUP ALL DISMOUNT;
```

The following SQL statement mounts disk group dgroup1:

```
ALTER DISKGROUP dgroup1 MOUNT;
```

Mounting Disk Groups Using the FORCE Option

For normal and high redundancy disk groups, you can use the FORCE option of the ALTER DISKGROUP statement's MOUNT clause to mount disk groups if there are sufficient ASM disks available. The disk group mount succeeds if ASM finds at least one complete set of extents in a disk group. If ASM determines that one or more disks are not available, then ASM moves those disks off line and drops the disks after the DISK_REPAIR_TIME expires.

In clustered ASM environments, if an ASM instance is not the first instance to mount the disk group, then using the MOUNT FORCE statement fails. This is because the disks have been accessed by another instance and the disks are not locally accessible.

Use the FORCE option as in the following example where *disk_group_name* is the name of the disk group that you want to force mount:

```
ALTER DISKGROUP disk_group_name MOUNT FORCE
```

> **See Also:** The *Oracle Database SQL Language Reference* for additional information about the ALTER DISKGROUP statement and the FORCE option

Checking the Internal Consistency of Disk Group Metadata

You can check the internal consistency of disk group metadata using the ALTER DISKGROUP statement with the CHECK keyword. You can use this statement to check specific files in a disk group, specific disks or all disks in a disk group, or specific failure groups within a disk group. The disk group must be mounted to perform these checks.

By default, the CHECK DISK GROUP clause verifies all of the metadata directories. ASM displays summary errors and writes the details about the errors in an alert log. The CHECK keyword performs the following operations:

- Verifies the consistency of the disk
- Cross checks all of 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

The REPAIR | NOREPAIR clause specifies whether ASM should attempt to repair errors that are found during the check. The default is NOREPAIR. Use the NOREPAIR clause to receive alerts about inconsistencies and to suppress ASM from resolving the errors automatically. The following example statement checks for consistency in the metadata for all disks in the dgroup1 disk group:

```
ALTER DISKGROUP dgroup1 CHECK ALL;
```

> **See Also:** The *Oracle Database SQL Language Reference* for additional information about the CHECK clause syntax

Dropping Disk Groups

The DROP DISKGROUP statement enables you to delete an ASM disk group and optionally, all of 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 provides syntactic consistency and prevents you from dropping the disk group if it has any contents

The ASM instance must be started and the disk group must be mounted with none of the disk group files open, in order for the DROP DISKGROUP statement to succeed. The statement does not return until the disk group has been dropped.

When you drop a disk group, ASM dismounts the disk group and removes the disk group name from the ASM_DISKGROUPS initialization parameter if a server parameter file is being used. If a text initialization parameter file is being used, and the disk group is mentioned in the ASM_DISKGROUPS initialization parameter, then you must remove the disk group name from the ASM_DISKGROUPS initialization parameter before the next time that you shut down and restart the ASM instance.

The following statement deletes dgroup1:

```
DROP DISKGROUP dgroup1;
```

After ensuring that none of the files contained in dgroup1 are open, ASM rewrites the header of each disk in the disk group to remove ASM formatting information. The statement does not specify INCLUDING CONTENTS, so the drop operation will fail if the disk group contains any files.

If you cannot mount a disk group but need to drop it, you can use the FORCE option of the DROP DISKGROUP statement. This command enables you to remove the headers on disks that belong to a disk group that cannot be mounted by any ASM instances as in the following example where *disk_group_name* is the name of the disk group:

```
DROP DISKGROUP disk_group_name FORCE
```

The disk group on which you perform this operation should not be mounted anywhere in the cluster. When you use the FORCE option, the ASM instance does not attempt to verify that a disk group is being used by another ASM instance in the same storage subsystem.

Note: Only use the FORCE option with extreme caution.

You can also drop a disk group with Oracle Enterprise Manager. See "Dropping Disk Groups" on page 6-10.

Using Views to Obtain ASM Information

You can use the views in Table 4–5 to obtain information about ASM:

Table 4–5 ASM Dynamic Views

View	Description
V$ASM_ALIAS	In an ASM instance, contains one row for every alias present in every disk group mounted by the ASM instance. In a DB instance, contains no rows.
V$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. Note that attributes are only displayed for disk groups where COMPATIBLE.ASM is set to 11.1 or higher.
V$ASM_CLIENT	In an ASM instance, identifies databases using disk groups managed by the ASM instance. In a DB instance, contains information about the ASM instance if the database has any open ASM files.
V$ASM_DISK	In an ASM instance, contains one row for every disk discovered by the ASM instance, including disks that are not part of any disk group. In a DB instance, contains rows only for disks in the disk groups in use by that DB instance. This view performs disk discovery every time it is queried.
V$ASM_DISK_IOSTAT	Displays information about disk I/O statistics for each ASM client. In a DB instance, only the rows for that instance are shown.
V$ASM_DISK_STAT	In an ASM instance, contains the same columns as V$ASM_DISK, but to reduce overhead, does not perform a discovery when it is queried. It does not return information about any disks that are new to the storage system. For the most accurate data, use V$ASM_DISK instead.
V$ASM_DISKGROUP	In an ASM instance, describes a disk group (number, name, size related info, state, and redundancy type). In a DB instance, contains one row for every ASM disk group mounted by the local ASM instance. This view performs disk discovery every time it is queried.

Table 4–5 (Cont.) ASM Dynamic Views

View	Description
V$ASM_DISKGROUP_STAT	In an ASM instance, contains the same columns as V$ASM_DISKGROUP, but to reduce overhead, does not perform a discovery when it is queried. It does not return information about any disks that are new to the storage system. For the most accurate data, use V$ASM_DISKGROUP instead.
V$ASM_FILE	In an ASM instance, contains one row for every ASM file in every disk group mounted by the ASM instance. In a DB instance, contains no rows.
V$ASM_OPERATION	In an ASM instance, contains one row for every active ASM long running operation executing in the ASM instance. In a DB instance, contains no rows.
V$ASM_TEMPLATE	In an ASM or DB instance, contains one row for every template present in every disk group mounted by the ASM instance.

An example of the use of the V$ASM_ATTRIBUTE and V$ASM_DISKGROUP views is shown in Example 4–2. Note that COMPATIBLE.ASM must be set to 11.1 or higher for the disk group to display in the V$ASM_ATTRIBUTE view output.

Example 4–2 Viewing Disk Groups with Associated Attributes

```
SQL> SELECT dg.name AS diskgroup, SUBSTR(a.name,1,24) AS name,
     SUBSTR(a.value,1,24) AS value FROM V$ASM_DISKGROUP dg, V$ASM_ATTRIBUTE a
     WHERE dg.name = 'DGROUP1' AND dg.group_number = a.group_number;

DISKGROUP                    NAME                     VALUE
---------------------------- ------------------------ ------------------------
DGROUP1                      compatible.rdbms         11.1.0.0.0
DGROUP1                      compatible.asm           11.1.0.0.0
DGROUP1                      au_size                  1048576
DGROUP1                      disk_repair_time         3.6h
...
```

An example of the use of the V$ASM_DISK and V$ASM_DISKGROUP views is shown in Example 4–3. This example displays the disks associated with a disk group, plus the minimum ASM and database software versions required to use the disk group.

Example 4–3 Viewing Disk Groups with Associated Disks

```
SQL> SELECT dg.name AS diskgroup, SUBSTR(d.name,1,16) AS asmdisk,
     SUBSTR(dg.compatibility,1,12) AS asm_compat,
     SUBSTR(dg.database_compatibility,1,12) AS db_compat,
     FROM V$ASM_DISKGROUP dg, V$ASM_DISK d
     WHERE dg.name LIKE 'DGROUP%' AND dg.group_number = d.group_number;

DISKGROUP                      ASMDISK          ASM_COMPAT   DB_COMPAT
------------------------------ ---------------- ------------ ------------
DGROUP1                        DISK1A           11.1.0.0.0   11.0.0.0.0
DGROUP1                        DISK1B           11.1.0.0.0   11.0.0.0.0
DGROUP1                        DISK1C           11.1.0.0.0   11.0.0.0.0
...
DGROUP2                        DISK2A           11.1.0.0.0   10.1.0.0.0
DGROUP2                        DISK2B           11.1.0.0.0   10.1.0.0.0
DGROUP2                        DISK2C           11.1.0.0.0   10.1.0.0.0
...
```

An example of the use of the `V$ASM_CLIENT` and `V$ASM_DISKGROUP` views on an ASM instance is shown in Example 4–4. This example displays disk groups with information about the connected database client instances.

Example 4–4 Viewing Disk Groups with Associated Client Version Settings

```
SQL> SELECT dg.name AS diskgroup, SUBSTR(c.instance_name,1,12) AS instance,
    SUBSTR(c.db_name,1,12) AS dbname, SUBSTR(c.SOFTWARE_VERSION,1,12) AS software,
    SUBSTR(c.COMPATIBLE_VERSION,1,12) AS compatible
    FROM V$ASM_DISKGROUP dg, V$ASM_CLIENT c
    WHERE  dg.group_number = c.group_number;

DISKGROUP                      INSTANCE     DBNAME   SOFTWARE     COMPATIBLE
------------------------------ ------------ -------- ------------ ------------
CONTROLFILE                    dblinux      dblinux  11.1.0.6.0   11.0.0.0.0
DATAFILE                       dblinux      dblinux  11.1.0.6.0   11.0.0.0.0
LOGFILE                        dblinux      dblinux  11.1.0.6.0   11.0.0.0.0
...
```

See Also: *Oracle Database Reference* for details on all of these dynamic performance views

5 Administering ASM Files, Directories, and Templates

This chapter describes how to administer files when you use the Automatic Storage Management (ASM) storage option. This appendix contains the following topics:

What Types of Files Does ASM Support?

ASM supports most file types required by the database. However, you cannot store some administrative file types on ASM disk groups. These include trace files, audit files, alert logs, export files, tar files, and core files.

Table 5–1 lists file types, indicates if they are supported, and lists the system default template that provides the attributes for file creation. Some of the file types shown in the table are related to specific products or features, and are not discussed in this book.

Table 5–1 File Types Supported by Automatic Storage Management

File Type	Supported	Default Templates
Control files	yes	`CONTROLFILE`
Datafiles	yes	`DATAFILE`
Redo log files	yes	`ONLINELOG`
Archive log files	yes	`ARCHIVELOG`
Trace files	no	n/a
Temporary files	yes	`TEMPFILE`
Datafile backup pieces	yes	`BACKUPSET`
Datafile incremental backup pieces	yes	`BACKUPSET`
Archive log backup piece	yes	`BACKUPSET`

Table 5–1 (Cont.) File Types Supported by Automatic Storage Management

File Type	Supported	Default Templates
Datafile copy	yes	`DATAFILE`
Persistent initialization parameter file (SPFILE)	yes	`PARAMETERFILE`
Disaster recovery configurations	yes	`DATAGUARDCONFIG`
Flashback logs	yes	`FLASHBACK`
Change tracking file	yes	`CHANGETRACKING`
Data Pump dumpset	yes	`DUMPSET`
Automatically generated control file backup	yes	`AUTOBACKUP`
Cross-platform transportable datafiles	yes	`XTRANSPORT`
Operating system files	no	n/a

See Also: "Managing Disk Group Templates" on page 5-15 for a description of the system default templates

About ASM Filenames

Every file created in ASM gets a system-generated filename, known as a fully qualified filename. The fully qualified filename represents a complete path name in the ASM file system. An example of a fully qualified filename is:

```
+dgroup2/sample/controlfile/Current.256.541956473
```

You can use the fully qualified filename to reference (read or retrieve) an ASM file. You can also use other abbreviated filename formats, such as an alias ASM filename described in "Alias ASM Filenames" on page 5-5, to reference an ASM file.

ASM generates a fully qualified filename upon any request to create a file. A creation request does not or cannot specify a fully qualified filename. Instead, it uses a simpler syntax to specify a file, such as an alias or just a disk group name. ASM then creates the file, placing it in the correct ASM *path* according to file type, and then assigns an appropriate fully qualified filename. If you specify an alias in the creation request, ASM also creates the alias so that it references the fully qualified filename.

ASM file creation requests are either single file creation requests or multiple file creation request.

Note: You can find the generated name in database views displaying Oracle file names, such as `V$DATAFILE` and `V$LOGFILE`. You can use this name, or an abbreviated form of it, if you later need to reference an ASM file in a SQL statement. Like other Oracle database filenames, ASM filenames are kept in the control file and the RMAN catalog.

Note: fully qualified and numeric filenames can be used in single-file create if you specify the `REUSE` keyword, as described in "Using ASM Filenames in SQL Statements" on page 5-9.

This sections contains the following topics:

- Single File Creation Request

- Multiple File Creation Request
- Fully Qualified ASM Filename
- Numeric ASM Filename
- Alias ASM Filenames
- Alias ASM Filename with Template
- Incomplete ASM Filename
- Incomplete ASM Filename with Template

Table 5–2 specifies the valid contexts for each filename form, and if the form is used for file creation, whether the created file is an Oracle Managed Files (OMF).

Table 5–2 Valid Contexts for the ASM Filename Forms

	Valid Context			OMF
Filename Form	**Reference**	**Single-File Creation**	**Multiple File Creation**	**Created as OMF?**
fully qualified filename	Yes	No	No	
Numeric filename	Yes	No	No	
Alias filename	Yes	Yes	No	No
Alias with template filename	No	Yes	No	No
Incomplete filename	No	Yes	Yes	Yes
Incomplete filename with template	No	Yes	Yes	Yes

Single File Creation Request

A single file creation request is a request to create a single file, such as a datafile or a control file. The form of the ASM filename in this type of request is either an alias (such as `+dgroup2/control/ctl.f`) or a disk group name preceded by a plus sign. You use the alias or disk group name where a filename is called for in a statement, such as `CREATE TABLESPACE` or `CREATE CONTROLFILE`.

> **Note:** '/ ' and '\' are interchangeable in filenames. Filenames are case insensitive, but case retentive.

Multiple File Creation Request

A multiple file creation request is a request that can occur multiple times to create an ASM file. For example, if you assign a value to the initialization parameter `DB_CREATE_FILE_DEST`, you can issue a `CREATE TABLESPACE` statement (without a filename specification) multiple times. Each time, ASM creates a different unique datafile name.

One form of the ASM filename to use in this type of request is an incomplete filename, which is just a disk group name preceded by a plus sign. In this case, you set `DB_CREATE_FILE_DEST` to an incomplete filename (for example, `+dgroup2`), and whenever a command is executed that must create a database file in `DB_CREATE_FILE_DEST`, the file is created in the designated disk group and assigned a unique fully qualified name. You can use an incomplete filename in other `*_DEST` initialization parameters.

Fully Qualified ASM Filename

This form of ASM filename can be used for referencing existing ASM files. It is the filename that ASM always automatically generates when an ASM file is created.

A fully qualified filename 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 directory of the ASM file system, similar to the slash (/) on UNIX or Linux computers.
- *dbname* is the `DB_UNIQUE_NAME` of the database to which the file belongs.
- *file_type* is the Oracle file type and can be one of the file types shown in Table 5–3.
- *file_type_tag* is type specific information about the file and can be one of the tags shown in Table 5–3.
- *file.incarnation* is the file/incarnation pair, used to ensure uniqueness.

An example of a fully qualified ASM filename is:

```
+dgroup2/sample/controlfile/Current.256.541956473
```

Table 5–3 Oracle File Types and Automatic Storage Management File Type Tags

Automatic Storage Management *file_type*	Description	Automatic Storage Management *file_type_tag*	Comments
CONTROLFILE	Control files and backup control files	Current Backup	--
DATAFILE	Datafiles and datafile copies	*tsname*	Tablespace into which the file is added
ONLINELOG	Online logs	group_*group#*	--
ARCHIVELOG	Archive logs	thread_*thread#*_seq_*sequence#*	--
TEMPFILE	Tempfiles	*tsname*	Tablespace into which the file is added
BACKUPSET	Datafile and archive log backup pieces; datafile incremental backup pieces	*hasspfile_timestamp*	*hasspfile* can take one of two values: *s* indicates that the backup set includes the spfile; *n* indicates that the backup set does not include the spfile.
PARAMETERFILE	Persistent parameter files	spfile	
DAATAGUARDCONFIG	Data Guard configuration file	*db_unique_name*	Data Guard attempts to use the service provider name if it is set. Otherwise the tag defaults to *DRCname*.
FLASHBACK	Flashback logs	log_*log#*	--
CHANGETRACKING	Block change tracking data	ctf	Used during incremental backups

Table 5–3 (Cont.) Oracle File Types and Automatic Storage Management File Type Tags

Automatic Storage Management *file_type*	Description	Automatic Storage Management *file_type_tag*	Comments
`DUMPSET`	Data Pump dumpset	`user_`*`obj#_file#`*	Dump set files encode the user name, the job number that created the dump set, and the file number as part of the tag.
`XTRANSPORT`	Datafile convert	*`tsname`*	--
`AUTOBACKUP`	Automatic backup files	*`hasspfile_timestamp`*	*`hasspfile`* can take one of two values: *`s`* indicates that the backup set includes the `spfile`; *`n`* indicates that the backup set does not include the `spfile`.

Numeric ASM Filename

The numeric ASM filename can be used for referencing existing files. It is derived from the fully qualified ASM filename and takes the form:

```
+group.file.incarnation
```

Numeric ASM filenames can be used in any interface that requires an existing file name.

An example of a numeric ASM filename is:

```
+dgroup2.257.541956473
```

Alias ASM Filenames

Alias ASM filenames, otherwise known as 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 filenames 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. You cannot create an alias at the root level (+), however.

When you create an ASM file with an alias filename, the file is created with a fully qualified name, and the alias filename is additionally created. You can then access the file with either name.

Alias ASM filenames are distinguished from fully qualified filenames or numeric filenames 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. Examples of ASM alias filenames are:

```
+dgroup1/myfiles/control_file1
+dgroup2/mydir/second.dbf
```

Oracle Database references database files by their alias filenames, but only if you create the database files with aliases. If you create database files without aliases and then add aliases later, the database references the files by their fully qualified filenames. The following are examples of how the database uses alias filenames:

- Alias filenames appear in V$ views. For example, if you create a tablespace and use an alias filename for the datafile, the `V$DATAFILE` view shows the alias filename.

- When a control file points to datafiles and online redo log files, it can use alias filenames.
- The CONTROL_FILES initialization parameter can use the alias filenames of the control files. The Database Configuration Assistant (DBCA) creates control files with alias filenames.

Note: Files created using an alias filename are not considered Oracle Managed Files and might require manual deletion in the future if they are no longer needed.

See Also:

- "Managing Alias Names for ASM Filenames" on page 5-9

Creating a Tablespace in ASM: Using a Datafile with an Alias Name

The following statement creates an undo tablespace with a datafile that has an alias name, and with attributes that are set by the user-defined template my_undo_template. This example assumes that a directory has been created in disk group dgroup3 to contain the alias name and that the user-defined template exists. Because an alias is used to create the datafile, the file is not an Oracle Managed Files (OMF) file and the file is not be automatically deleted when the tablespace is dropped.

```
CREATE UNDO TABLESPACE myundo
     DATAFILE '+dgroup3(my_undo_template)/myfiles/my_undo_ts' SIZE 200M;
```

The following statement drops the file manually after the tablespace has been dropped:

```
ALTER DISKGROUP dgroup3 DROP FILE '+dgroup3/myfiles/my_undo_ts';
```

Alias ASM Filename with Template

An alias ASM filename with template is used only for ASM file creation operations. It has the following format:

```
+dgroup(template_name)/alias
```

Alias filenames with template behave identically to alias filenames. The only difference is that a file created with an alias filename with template 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.

The creation and maintenance of ASM templates is discussed in "Managing Disk Group Templates" on page 5-15.

An example of an alias ASM filename with template is:

```
+dgroup1(my_template)/config1
```

Explicitly specifying a template name, as in this example, overrides the system default template for the type of file being created.

Note: Files created using an alias filename with template are not considered Oracle Managed Files and might require manual deletion in the future if they are no longer needed.

Incomplete ASM Filename

Incomplete ASM filenames are used only for file creation operations and are used for both single and multiple file creation. They consist only of the disk group name. ASM uses a system default template to determine the ASM file mirroring and striping attributes. The system template that is used is determined by the file type that is being created. For example, if you are creating a datafile for a tablespace, the `datafile` template is used.

An example of using an incomplete ASM filename is setting the `DB_CREATE_FILE_DEST` initialization parameter to:

```
+dgroup1
```

With this setting, every time you create a tablespace, a datafile is created in the disk group `dgroup1`, and each datafile is assigned a different fully qualified name. See "Creating ASM Files Using a Default Disk Group Specification" on page 5-7 for more information.

Incomplete ASM Filename with Template

Incomplete ASM filenames with templates are used only for file creation operations and are used for both single and multiple file creation. They consist of the disk group name followed by the template name in parentheses. When you explicitly specify a template in a file name, ASM uses the specified template instead of the default template for that file type to determine mirroring and striping attributes for the file.

An example of using an incomplete ASM filename with template is setting the `DB_CREATE_FILE_DEST` initialization parameter to:

```
+dgroup1(my_template)
```

Creating and Referencing ASM Files in the Database

An ASM file is an Oracle Managed File unless you created the file using an alias. Any OMF is automatically deleted when it is no longer needed. An ASM file is deleted if the creation fails.

This section contains the following topics:

- Creating ASM Files Using a Default Disk Group Specification
- Using ASM Filenames in SQL Statements

Creating ASM Files Using a Default Disk Group Specification

Using the Oracle Managed Files feature for operating system files, you can specify a directory as the default location for the creation of datafiles, temporary files, redo log files, and control files. Using the Oracle Managed Files feature for ASM, you can specify a disk group, in the form of an incomplete ASM filename, as the default location for creation of these files, and additional types of files, including archived log files. As for operating system files, the name of the default disk group is stored in an initialization parameter and is used whenever a file specification (for example, `DATAFILE` clause) is not explicitly specified during file creation.

The initialization parameters in Table 5–4 accept the multiple file creation context form of ASM filenames as a destination:

Table 5–4 Multifile ASM Filename Initialization Parameters

Initialization Parameter	Description
DB_CREATE_FILE_DEST	Specifies the default disk group location in which to create: ■ Datafiles ■ Tempfiles If DB_CREATE_ONLINE_LOG_DEST_*n* is not specified, then also specifies the default disk group for: ■ Redo log files ■ Control file
DB_CREATE_ONLINE_LOG_DEST_*n*	Specifies the default disk group location in which to create: ■ Redo log files ■ Control files
DB_RECOVERY_FILE_DEST	If this parameter is specified and DB_CREATE_ONLINE_LOG_DEST_*n* and CONTROL_FILES are not specified, then this parameter specifies a default disk group for a flash recovery area that contains a copy of: ■ Control file ■ Redo log files If no local archive destination is specified, then this parameter implicitly sets LOG_ARCHIVE_DEST_10 to the USE_DB_RECOVERY_FILE_DEST value.
CONTROL_FILES	Specifies a disk group in which to create control files.

The initialization parameters in Table 5–5 accept the multiple file creation context form of the ASM filenames and ASM directory names as a destination:

Table 5–5 Multifile ASM Filename and Directory Name Initialization Parameters

Initialization Parameter	Description
LOG_ARCHIVE_DEST_*n*	Specifies a default disk group or ASM directory as destination for archiving redo log files
LOG_ARCHIVE_DEST	Optional parameter to use to specify a default disk group or ASM directory as destination for archiving redo log files. Use when specifying only one destination.
STANDBY_ARCHIVE_DEST	Relevant only for a standby database in managed recovery mode. It specifies a default disk group or ASM directory that is the location of archive logs arriving from a primary database. Not discussed in this book. This parameter has been deprecated. See *Oracle Data Guard Concepts and Administration*.

The following example illustrates how an ASM file, in this case a datafile, might be created in a default disk group.

Creating a Datafile Using a Default Disk Group: Example

Assume the following initialization parameter setting:

```
DB_CREATE_FILE_DEST = '+dgroup1'
```

The following statement creates tablespace tspace1.

```
CREATE TABLESPACE tspace1;
```

ASM automatically creates and manages the datafile for tspace1 on ASM disks in the disk group dgroup1. File extents are stored using the attributes defined by the default template for a datafile.

Using ASM Filenames in SQL Statements

You can specify ASM filenames in the file specification clause of your SQL statements. If you are creating a file for the first time, use the creation form of an ASM filename. If the ASM file already exists, you must use the reference context form of the filename, and if you are trying to re-create the file, you must add the REUSE keyword. The space will be reused for the new file. This usage might occur when, for example, trying to re-create a control file, as shown in "Creating Control Files in ASM" on page A-3.

If a reference context form is used with the REUSE keyword and the file does not exist, an error results.

Partially created files resulting from system errors are automatically deleted.

Using an ASM Filename in a SQL Statement: Example

The following is an example of specifying an ASM filename in a SQL statement. In this case, it is used in the file creation context:

```
CREATE TABLESPACE  tspace2 DATAFILE '+dgroup2' SIZE 200M AUTOEXTEND ON;
```

The tablespace tspace2 is created and is comprised of one datafile of size 200 MB contained in the disk group dgroup2. The datafile is set to auto-extensible with an unlimited maximum size. An AUTOEXTEND clause can be used to override this default.

Managing Alias Names for ASM Filenames

Alias names, or aliases, are intended to provide a more user-friendly means of referring to ASM files, rather than using the system-generated filenames.

You can create an alias for a file when you create it in the database, or you can add an alias to an existing file using the ADD ALIAS clause of the ALTER DISKGROUP statement. You can create an alias in any system-generated or user-created ASM directory. You cannot create an alias at the root level (+), however.

For more information about creating aliases with ASMCMD, refer to "mkalias Command" on page 7-18.

This section contains the following sections:

- Adding an Alias Name for an ASM Filename
- Renaming an Alias Name for an ASM Filename
- Dropping an Alias Name for an ASM Filename
- Dropping Files and Associated Aliases from a Disk Group

Adding an Alias Name for an ASM Filename

Use the ADD ALIAS clause of the ALTER DISKGROUP statement to create an alias name for an ASM filename. The alias name must consist of the full directory path and the alias itself.

Example 1: Adding an Alias Name for an ASM Filename

The following statement adds a new alias name for a system-generated file name:

```
ALTER DISKGROUP dgroup1 ADD ALIAS '+dgroup1/mydir/second.dbf'
     FOR '+dgroup1/sample/datafile/mytable.342.3';
```

Example 2: Adding an Alias Name for an ASM Filename

This statement illustrates another means of specifying the ASM filename for which the alias is to be created. It uses the numeric form of the ASM filename, which is an abbreviated and derived form of the system-generated filename.

```
ALTER DISKGROUP dgroup1 ADD ALIAS '+dgroup1/mydir/second.dbf'
     FOR '+dgroup1.342.3';
```

Renaming an Alias Name for an ASM Filename

Use the RENAME ALIAS clause of the ALTER DISKGROUP statement to rename an alias for an ASM filename. The old and the new alias names must consist of the full directory paths of the alias names.

Example: Renaming an Alias Name for an ASM Filename

The following statement renames an alias:

```
ALTER DISKGROUP dgroup1 RENAME ALIAS '+dgroup1/mydir/datafile.dbf'
     TO '+dgroup1/payroll/compensation.dbf';
```

Dropping an Alias Name for an ASM Filename

Use the DROP ALIAS clause of the ALTER DISKGROUP statement to drop an alias for an ASM filename. The alias name must consist of the full directory path and the alias itself. The underlying file to which the alias refers is unchanged.

Example 1: Dropping an Alias Name for an ASM Filename

The following statement drops an alias:

```
ALTER DISKGROUP dgroup1 DROP ALIAS '+dgroup1/payroll/compensation.dbf';
```

Example 2: Dropping an Alias Name for an ASM Filename

The following statement will fail because it attempts to drop a system-generated filename. This is not allowed:

```
ALTER DISKGROUP dgroup1
     DROP ALIAS '+dgroup1/sample/datafile/mytable.342.3';
```

Dropping Files and Associated Aliases from a Disk Group

You can delete ASM files and their associated aliases from a disk group using the DROP FILE clause of the ALTER DISKGROUP statement. You must use a fully qualified filename, a numeric filename, or an alias name when specifying the file that you want to delete. The underlying file on the file system is not dropped when the alias is dropped.

Some reasons why you might need to delete files include:

- A file created using aliases is not an Oracle Managed File. Consequently, it is not automatically deleted.
- A point-in-time-recovery of a database might restore the database to a time before a tablespace was created. The restore does not delete the tablespace, but there is no

reference to the tablespace or its datafile in the restored database. You could manually delete the datafile.

Example 1: Dropping Files and Associated Aliases from a Disk Group

In the following example, the alias name for the file is used to delete both the file and the alias from a disk group:

```
ALTER DISKGROUP dgroup1 DROP FILE '+dgroup1/payroll/compensation.dbf';
```

Example 2: Dropping Files and Associated Aliases from a Disk Group

In the following example, the system-generated filename is used to drop the file and any associated alias:

```
ALTER DISKGROUP dgroup1
     DROP FILE '+dgroup1/sample/datafile/mytable.342.372642';
```

Accessing ASM Files with the XML DB Virtual Folder

ASM files and directories can be accessed through a virtual folder in the 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.

A typical use for this capability might be to view /sys/asm as a Web Folder in a graphical user interface (with the WebDAV protocol), and then copy a Data Pump dumpset from an ASM disk group to an operating system file system by dragging and dropping.

You must log in as a user other than SYS and you must have been granted the DBA role to access /sys/asm with XML DB protocols.

Note: The FTP protocol is initially disabled for a new XML DB installation. To enable it, you must set the FTP port to a nonzero value. The easiest way to do this is with the catxdbdbca.sql script. This script takes two arguments. The first is the FTP port number, and the second is the HTTP/WebDAV port number. The following example configures the FTP port number to 7787, and the HTTP/WebDAV port number to 8080:

```
SQL> @?/rdbms/admin/catxdbdbca.sql 7787 8080
```

Another way to set these port numbers is with the XDB Configuration page in Enterprise Manager.

See Also:

- *Oracle XML DB Developer's Guide* for information about Oracle XML DB, including additional ways to configure port numbers for the XML DB protocol servers
- *Oracle Database PL/SQL Packages and Types Reference* for information about the DBMS_XDB package

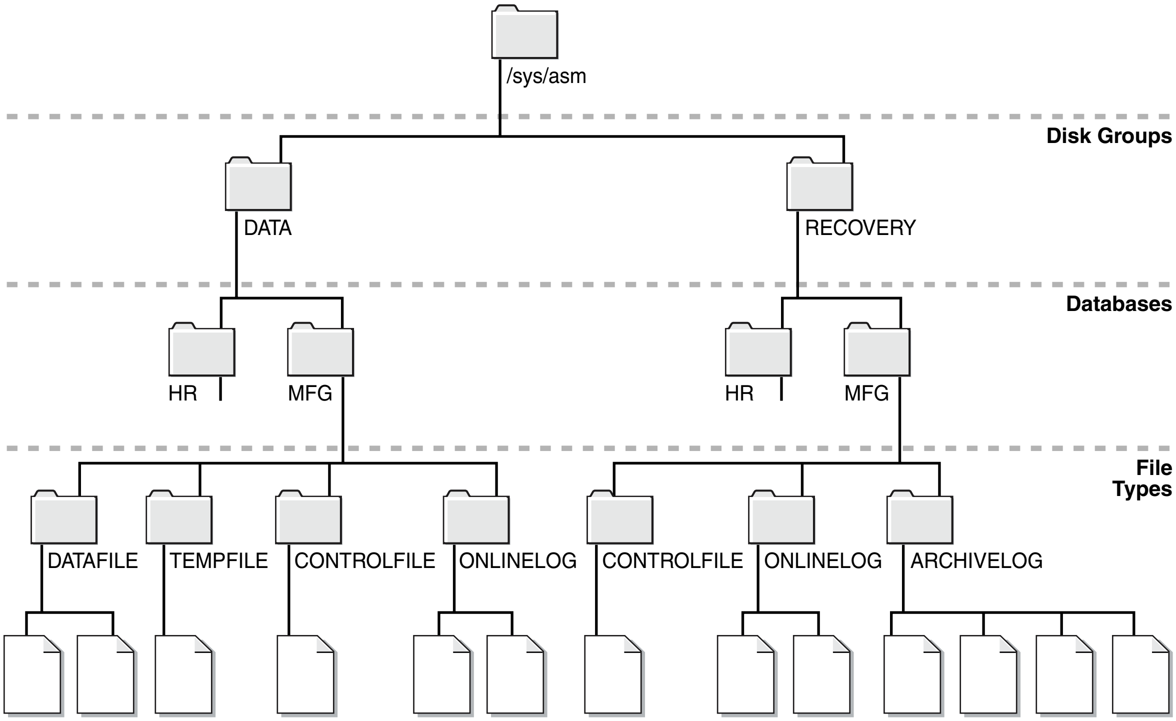
Inside /sys/asm

The ASM virtual folder is created by default during XML DB installation. If the database is not configured to use ASM, the folder is empty and no operations are permitted on it.

The ASM virtual folder contains folders and subfolders that follow the hierarchy defined by the structure of an ASM fully qualified file name. Figure 5–1 show this hierarchy, which for simplicity, excludes directories created for aliases.

The folder `/sys/asm` 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 file type folders, which contain the ASM files.

Figure 5–1 Hierarchal Structure of ASM Folders



Restrictions

The following are usage restrictions on `/sys/asm`:

- You cannot create hard links to existing ASM files or directories with APIs such as DBMS_XDB.LINK.
- You cannot rename (move) an ASM file to another disk group or to a directory outside ASM.

You can use the directory `/sys/asm` for storing the names of disk groups. You cannot store other files in this directory. Within the disk group directories under `/sys/asm`, such as `/sys/asm/DATA`, you can only store database files in these sub-directories. ASM rejects attempts to store non-database files in these directories.

Sample FTP Session

In the following sample FTP session, the disk groups are DATA and RECOVERY, the database name is MFG, and dbs is a directory that was created for aliases. All files in /sys/asm are binary.

```
ftp> open myhost 7777
ftp> user system
ftp> passwd dba
ftp> cd /sys/asm
ftp> ls
DATA
RECOVERY
ftp> cd DATA
ftp> ls
dbs
MFG
ftp> cd dbs
ftp> ls
t_dbl.f
t_axl.f
ftp> binary
ftp> get t_dbl.f t_axl.f
ftp> put t_db2.f
```

Using DBMS_FILE Transfer Utility for ASM

The DBMS_FILE_TRANSFER package provides procedures to copy ASM files within a database or to 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.

> **See Also:** *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS_FILE_TRANSFER package

Managing Disk Group Directories

ASM disk groups contain a system-generated hierarchical directory structure for storing ASM files. The system-generated filename that ASM assigns to each file represents a path in this directory hierarchy. The following is an example of a system-generated filename:

```
+dgroup2/sample/CONTROLFILE/Current.256.541956473
```

The plus sign represents the root of the ASM file system. The dgroup2 directory is the parent directory for all files in the dgroup2 disk group. The sample directory is the parent directory for all files in the sample database, and the CONTROLFILE directory contains all control files for the sample 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.

This section describes how to use the ALTER DISKGROUP statement to create a directory structure for aliases. It also describes how you can rename a directory or drop a directory. This section contains the following topics:

- Creating a New Directory
- Renaming a Directory
- Dropping a Directory

Creating a New Directory

Use the ADD DIRECTORY clause of the ALTER DISKGROUP statement to create a hierarchical directory structure for alias names for ASM files. Use the slash character (/) to separate components of the directory path. The directory path must start with the disk group name, preceded by a plus sign (+), followed by any subdirectory names of your choice.

The parent directory must exist before attempting to create a subdirectory or alias in that directory.

Example 1: Creating a New Directory

The following statement creates a hierarchical directory for disk group dgroup1, which can contain, for example, the alias name +dgroup1/mydir/control_file1:

```
ALTER DISKGROUP dgroup1 ADD DIRECTORY '+dgroup1/mydir';
```

Example 2: Creating a New Directory

Assuming no subdirectory exists under the directory +dgoup1/mydir, the following statement fails:

```
ALTER DISKGROUP dgroup1
     ADD DIRECTORY '+dgroup1/mydir/first_dir/second_dir';
```

Renaming a Directory

The RENAME DIRECTORY clause of the ALTER DISKGROUP statement enables you to rename a directory. System created directories (those containing system-generated names) cannot be renamed.

Example: Renaming a Directory

The following statement renames a directory:

```
ALTER DISKGROUP dgroup1 RENAME DIRECTORY '+dgroup1/mydir'
     TO '+dgroup1/yourdir';
```

Dropping a Directory

You can delete a directory using the DROP DIRECTORY clause of the ALTER DISKGROUP statement. You cannot drop a system created directory. You cannot drop a directory containing alias names unless you also specify the FORCE clause.

Example: Dropping a Directory

This statement deletes a directory along with its contents:

```
ALTER DISKGROUP dgroup1 DROP DIRECTORY '+dgroup1/yourdir' FORCE;
```

Managing Disk Group Templates

This section describes how to manage disk group templates under the following topics:

- Template Attributes
- Adding Templates to a Disk Group
- Modifying a Disk Group Template
- Dropping Templates from a Disk Group
- Creating Tablespaces in ASM: Specifying Redundancy and Striping with Templates

Templates are used to set redundancy (mirroring) and striping attributes of files 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 ASM disks. Default template settings depend on the disk group type. The default template for datafiles for a normal redundancy disk group sets two-way mirroring, while the corresponding default template in a high redundancy disk group sets three-way mirroring. You can modify these default templates.

For example, default redundancy for the online redo log files (ONLINELOG template) for a normal redundancy disk group is MIRROR. In Example 4–1 on page 4-3, this means that when one copy of a redo log file extent is written to a disk in failure group controller1, a mirrored copy of the file extent is written to a disk in failure group controller2. To support the default mirroring of a normal redundancy disk group, at least two failure groups must be defined.

Table 5–8 on page 5-16 lists the default templates and the attributes that are associated to matching files. As the table shows, the initial redundancy value of each default template depends on the type of disk group that the template belongs to.

Note: The striping attribute of templates applies to all disk group types. This includes normal redundancy, high redundancy, and external redundancy disk group types. However, the mirroring attribute of templates applies only to normal redundancy disk groups, and is ignored for high-redundancy disk groups in which every file is always three-way mirrored. It is also 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.

Using clauses of the ALTER DISKGROUP statement, you can add new templates to a disk group, modify existing ones, or drop templates. The reason to add templates is to create the right combination of attributes to meet unique requirements. You can then reference a template name when creating a file, thereby assigning desired attributes based on an individual file rather than on the file type. The V$ASM_TEMPLATE view lists all of the templates known to the ASM instance.

Template Attributes

Table 5–6 shows the permitted striping attribute values and Table 5–7 shows the permitted redundancy values for ASM templates. These values correspond to the STRIPE and REDUND columns of V$ASM_TEMPLATE.

Table 5–6 Permitted Values for ASM Template Striping Attribute

Striping Attribute Value	Description
FINE	Striping in 128 KB chunks.
COARSE	Striping in 1 MB chunks.

Table 5–7 Permitted Values for ASM Template Redundancy Attribute

Redundancy Attribute Value	Resulting Mirroring in Normal Redundancy Disk Group	Resulting Mirroring in High Redundancy Disk Group	Resulting Mirroring in External Redundancy Disk Group
MIRROR	Two-way mirroring	Three-way mirroring	(Not allowed)
HIGH	Three-way mirroring	Three-way mirroring	(Not allowed)
UNPROTECTED	No mirroring	(Not allowed)	No mirroring

Table 5–8 ASM System Default Templates Attribute Settings

Template Name	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	FINE	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	FINE	MIRROR	HIGH	UNPROTECTED
CHANGETRACKING	COARSE	MIRROR	HIGH	UNPROTECTED
DUMPSET	COARSE	MIRROR	HIGH	UNPROTECTED
XTRANSPORT	COARSE	MIRROR	HIGH	UNPROTECTED
AUTOBACKUP	COARSE	MIRROR	HIGH	UNPROTECTED

Adding Templates to a Disk Group

To add a template to a disk group, use the ADD TEMPLATE clause of the ALTER DISKGROUP statement. You specify the name of the template, its redundancy attribute, and its striping attribute.

> **Note:** If the name of your new template is not one of the names listed in Table 5–8, then it is not used as a default template for database file types. To use the name, you must reference its name when creating a file.

The syntax of the ALTER DISKGROUP command for adding a template is as follows:

```
ALTER DISKGROUP disk_group_name ADD TEMPLATE template_name
  ATTRIBUTES ([{MIRROR|HIGH|UNPROTECTED}] [{FINE|COARSE}]);
```

Both types of attribute are optional. If no redundancy attribute is specified, 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. If you do not specify a striping attribute, then the value defaults to COARSE.

Example 1: Adding a Template to a Disk Group

The following statement creates a new template named reliable for the normal redundancy disk group dgroup2:

```
ALTER DISKGROUP dgroup2 ADD TEMPLATE reliable ATTRIBUTES (HIGH FINE);
```

Example 2: Adding a Template to a Disk Group

The following statement creates a new template named unreliable that specifies files are to be unprotected (no mirroring):

```
ALTER DISKGROUP dgroup2 ADD TEMPLATE unreliable ATTRIBUTES (UNPROTECTED);
```

> **Note:** Oracle discourages using unprotected files unless you have implemented hardware mirroring. The previous example is presented only to further illustrate how the attributes for templates are set.

> **See Also:** *Oracle Database SQL Language Reference* for more information about the ALTER DISKGROUP...ADD TEMPLATE command.

Modifying a Disk Group Template

The MODIFY TEMPLATE clause of the ALTER DISKGROUP statement enables you to modify the attribute specifications of an existing system default or user-defined disk group template. Only specified template properties are changed. Unspecified properties retain their current value. When you modify an existing template, only new files created by the template reflect the attribute changes. Existing files maintain their attributes.

Example: Modifying a Disk Group Template

The following example changes the striping attribute specification of the reliable template for disk group dgroup2:

```
ALTER DISKGROUP dgroup2 MODIFY TEMPLATE reliable
     ATTRIBUTES (COARSE);
```

Dropping Templates from a Disk Group

Use the `DROP TEMPLATE` clause of the `ALTER DISKGROUP` statement to drop one or more templates from a disk group. You can only drop templates that are user-defined; you cannot drop system default templates.

Example: Dropping a Template from a Disk Group

The following example drops the previously created template `unreliable` from `dgroup2`:

```
ALTER DISKGROUP dgroup2 DROP TEMPLATE unreliable;
```

Creating Tablespaces in ASM: Specifying Redundancy and Striping with Templates

Use the SQL `ALTER SYSTEM` and `CREATE TABLESPACE` statements to create a tablespace that uses a user-defined template to specify the redundancy and striping attributes of the datafile.

Example: Using a User-Defined Template to Specify Redundancy and Striping

The following example assumes that the template (`my_template`) has been defined.

```
ALTER SYSTEM SET DB_CREATE_FILE_DEST = '+dgroup1(my_template)';

CREATE TABLESPACE tspace3;
```

6

Administering ASM with Oracle Enterprise Manager

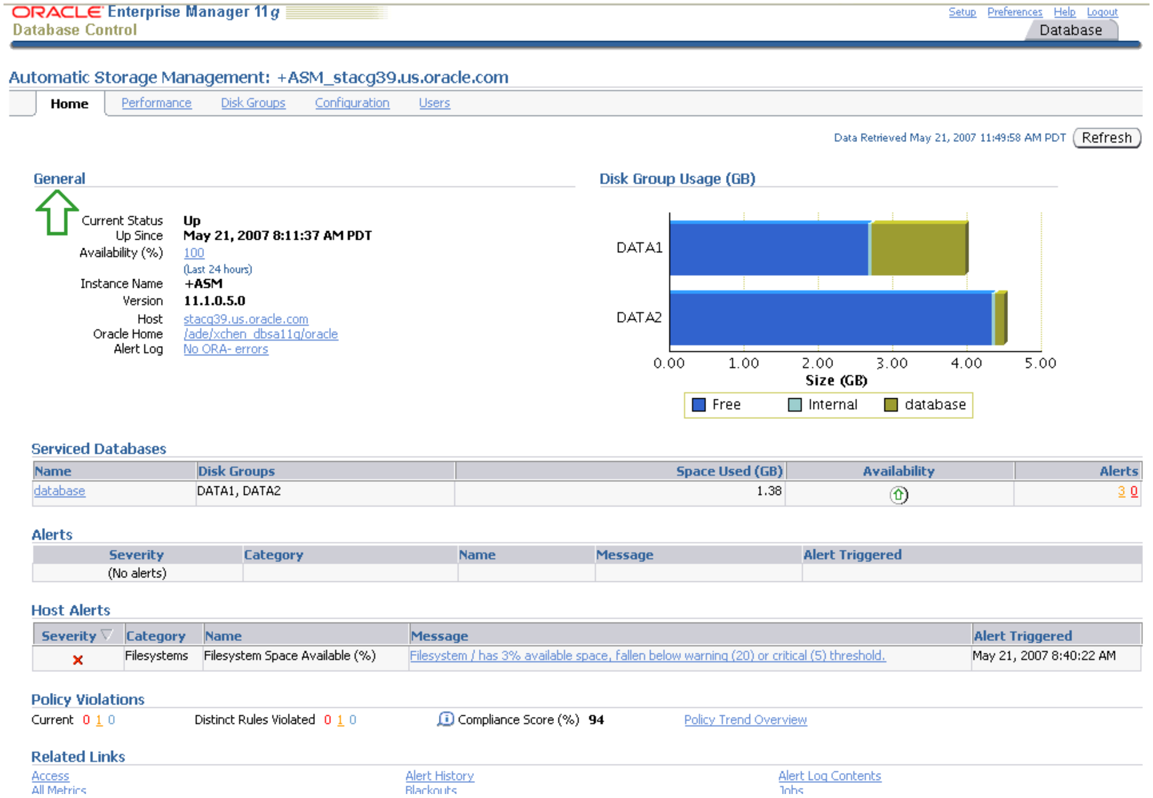
This chapter describes how to administer Automatic Storage Management (ASM) by using Oracle Enterprise Manager to provision and manage datafile storage. This chapter contains the following topics:

The Automatic Storage Management Home Page

All ASM administration tasks begin with the Automatic Storage Management Home page in Enterprise Manager, shown in Figure 6–1. The Automatic Storage Management Home page displays:

- The status of the ASM instance
- A chart that shows the used and free space of each disk group and disk group internal usage
- A list of databases that are serviced by the ASM instance
- Links to the ASM Performance, Disk Groups, Configuration, and Users pages

Figure 6–1 Automatic Storage Management Home Page



Automatic Storage Management: +ASM_stacg39.us.oracle.com

Data Retrieved May 21, 2007 11:49:58 AM PDT

General

Current Status	Up
Up Since	May 21, 2007 8:11:37 AM PDT
Availability (%)	100 (Last 24 hours)
Instance Name	+ASM
Version	11.1.0.5.0
Host	stacg39.us.oracle.com
Oracle Home	/ade/xchen_dbsa11g/oracle
Alert Log	No ORA- errors

Serviced Databases

Name	Disk Groups	Space Used (GB)	Availability	Alerts
database	DATA1, DATA2	1.38	⇧	3 0

Alerts

Severity	Category	Name	Message	Alert Triggered
(No alerts)				

Host Alerts

Severity ▽	Category	Name	Message	Alert Triggered
✗	Filesystems	Filesystem Space Available (%)	Filesystem / has 3% available space, fallen below warning (20) or critical (5) threshold.	May 21, 2007 8:40:22 AM

Policy Violations

Current 0 1 0 Distinct Rules Violated 0 1 0 Compliance Score (%) 94 Policy Trend Overview

Related Links

Access	Alert History	Alert Log Contents
All Metrics	Blackouts	Jobs

Accessing the ASM Home Page in Single-Instance Oracle Databases

To access the Automatic Storage Management Home page on a single-instance system:

1. Access the Database Home page by logging in as any database user.

   See Also: *Oracle Database 2 Day DBA* for more information

2. Under the heading labeled **General** on the left side of the Database Instance Home page as shown in Figure 6–1, click the link next to the last entry in the General column labeled **ASM**.

3. If prompted for ASM login credentials, then enter the user SYS, provide the SYS password that was set for the ASM instance upon installation, and connect as SYSASM. Enterprise Manager displays the Automatic Storage Management Home page similar to the page in Figure 6–1.

   See Also: "Authentication for Accessing ASM Instances" on page 3-17 for more information about authentication

Accessing the Automatic Storage Management Home Page in Oracle RAC Databases

To access the Automatic Storage Management Home page in an Oracle Real Application Clusters (Oracle RAC) environment:

1. Log in to Enterprise Manager Database Control on any node that is running the Oracle Management Service (OMS).

OMS is automatically started on the node on which Database Configuration Assistant (DBCA) was originally run to create the cluster database. Depending on your configuration, OMS might also be running on other nodes.

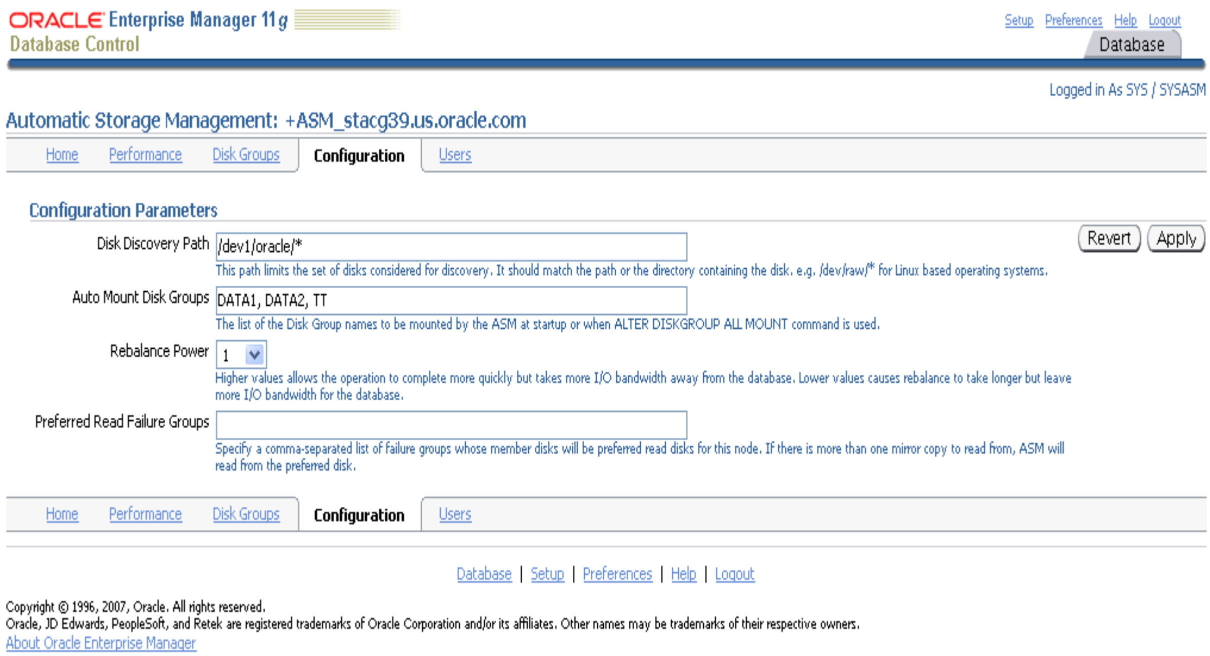
> **See Also:** *Oracle Database 2 Day + Real Application Clusters Guide* for more information

2. On the Cluster Database page, under the Instances heading, click the link for the ASM instance that you want to access.

Configuring ASM Initialization Parameters

You can configure ASM with the Enterprise Manager Configuration Parameters page similar to the one shown in Figure 6–2.

Figure 6–2 Automatic Storage Management Configuration Parameters Page


ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Database
Logged in As SYS / SYSASM
Home Performance Disk Groups Configuration Users
Revert
Apply
Home Performance Disk Groups Configuration Users
Database | Setup | Preferences | Help | Logout

Automatic Storage Management: +ASM_stacg39.us.oracle.com

Configuration Parameters

Disk Discovery Path: /dev1/oracle/*
This path limits the set of disks considered for discovery. It should match the path or the directory containing the disk. e.g. /dev/raw/* for Linux based operating systems.

Auto Mount Disk Groups: DATA1, DATA2, TT
The list of the Disk Group names to be mounted by the ASM at startup or when ALTER DISKGROUP ALL MOUNT command is used.

Rebalance Power: 1
Higher values allows the operation to complete more quickly but takes more I/O bandwidth away from the database. Lower values causes rebalance to take longer but leave more I/O bandwidth for the database.

Preferred Read Failure Groups:
Specify a comma-separated list of failure groups whose member disks will be preferred read disks for this node. If there is more than one mirror copy to read from, ASM will read from the preferred disk.

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

To configure the ASM instance:

1. Access the Automatic Storage Management Home page.

   See "The Automatic Storage Management Home Page" on page 6-1 for instructions.

2. Click the **Configuration** link at the top of the page to view the Configuration Parameters page.

3. Update the configuration options on the page:

   - Disk Discovery Path (`ASM_DISKSTRING` initialization parameter)

     Enter a string value. See "ASM_DISKSTRING" on page 3-5.

   - Auto Mount Disk Groups (`ASM_DISKGROUPS` initialization parameter)

     Enter a string value. See "ASM_DISKGROUPS" on page 3-5.

- Rebalance Power (`ASM_POWER_LIMIT` initialization parameter)

  Enter an integer value. See "ASM_POWER_LIMIT" on page 3-6.

- Preferred Read Failure Groups (Only in Oracle RAC environments) (`ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter)

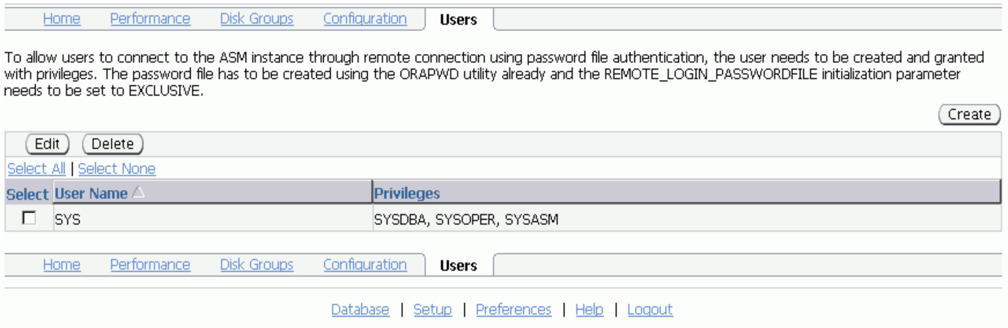
  Enter a comma-delimited string of values in which each string represents a failure group. This parameter is generally used only for clustered ASM instances. See "ASM_PREFERRED_READ_FAILURE_GROUPS" on page 3-6.

4. Click **Apply** to save your changes or click **Revert** to discard any changes made on the page.

Managing ASM Users

You can manage ASM users with the Enterprise Manager Users page, shown in Figure 6–3. Access this page by clicking the Users tab that is near the top center of the Automatic Storage Management Home page.

Figure 6–3 Automatic Storage Management Users Dialog



To allow users to connect to the ASM instance through remote connection using password file authentication, the user needs to be created and granted with privileges. The password file has to be created using the ORAPWD utility already and the REMOTE_LOGIN_PASSWORDFILE initialization parameter needs to be set to EXCLUSIVE.

Select	User Name △	Privileges
□	SYS	SYSDBA, SYSOPER, SYSASM

To manage ASM users:

1. Access the Automatic Storage Management Home page.

   See "The Automatic Storage Management Home Page" on page 6-1 for instructions.

2. Click the **Users** link at the top to view the Users property page.

3. Click **Create** to add a new ASM user. When you add users, Enterprise Manager displays a Create User page in which you must enter user login credentials that include the user name, password, and password confirmation. You can also grant privileges to the new user by selecting privileges in the Available Privileges column and clicking the right arrow buttons to move privileges to the Granted Privileges column, or move the privilege by double clicking it. The available privileges include SYSDBA, SYSOPER, and SYSASM. Click **OK** to create the user, **Cancel** to cancel the procedure, or **Show SQL** to view the SQL that Enterprise Manager uses to create the user.

4. To edit a user’s properties, click the box in the Select column for the user that you want to edit and click **Edit**. On the Edit User page, modify the password if needed. You can also alter the privileges that are assigned to the user by selecting the privilege and using the right and left arrow buttons to move the privilege from the Granted Privileges column to the Available Privileges column and vice versa. You can also move the privilege by double clicking it. Click **OK** to edit the user

properties, **Revert** to cancel the procedure, or **Show SQL** to view the SQL that Enterprise Manager uses to edit the user's properties.

5. To delete an ASM user, click the box in the Select column for the user that you want to delete and click **Delete**. On the confirmation dialog, click **Yes** to delete the user or **No** to stop the user deletion procedure.

> **See Also:** "Authentication for Accessing ASM Instances" on page 3-17 for more information about user authentication

Managing Disk Groups

This section describes how to manage disk groups under the following topics:

- Creating Disk Groups
- Adding Disks to Disk Groups
- Dropping Disks from Disk Groups
- Dropping Disk Groups
- Monitoring Disk Group Usage
- Bringing Disks Online and Offline
- Mounting and Dismounting Disk Groups
- Administering Advanced Disk Group Properties

Creating Disk Groups

You can create additional ASM disk groups to:

- Have disk groups with different redundancy levels (normal, high, or external), depending on availability requirements and storage system capabilities.
- Separate different classes of storage, such as SCSI drives and SATA drives, into different disk groups. Disks in a disk group should have similar size and performance characteristics.
- Store the flash recovery area in a separate disk group from the database.

To create a disk group:

1. Access the Automatic Storage Management Home page.

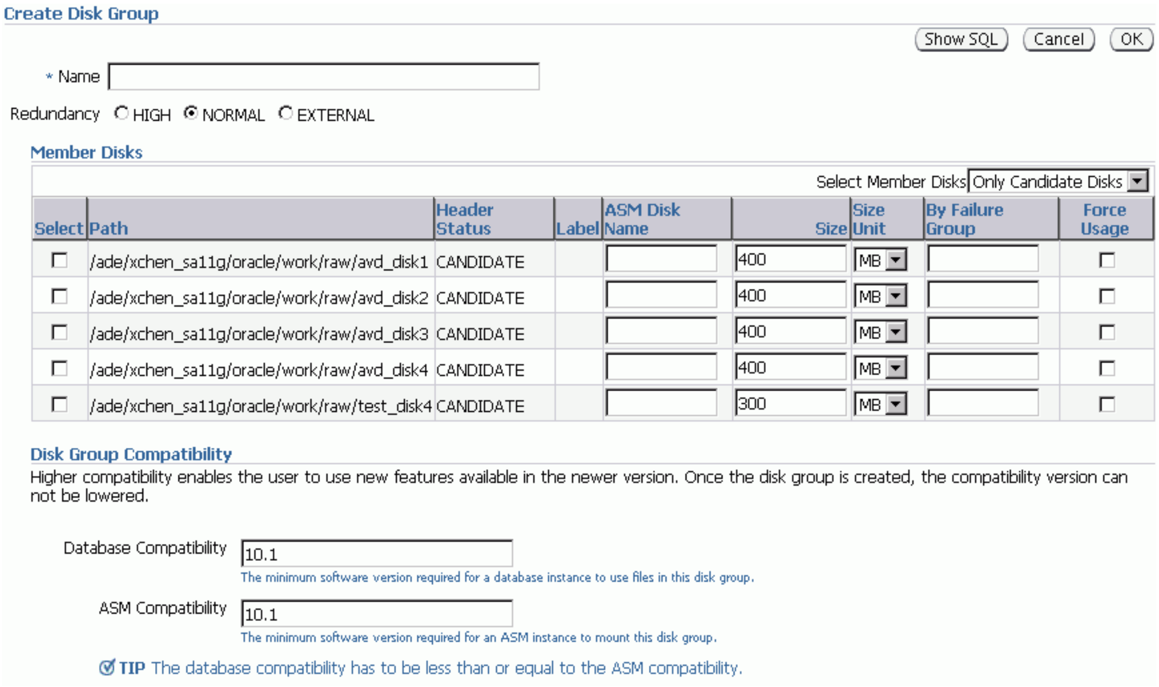
   See "The Automatic Storage Management Home Page" on page 6-1 for instructions.

2. Click the **Disk Groups** link at the top of the page to display the Disk Groups property page.

3. If you are not already logged in, then the Log In page appears and you should log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password for the ASM instance. This password was set up when the ASM instance was created.

4. Click **Create** and Enterprise Manager displays a Create Disk Group page similar to the page in Figure 6–4. The Create Disk Group page displays a list of ASM disks that are available to be added to a disk group. This includes disks with a header status of `CANDIDATE`, `PROVISIONED`, or `FORMER`, and so on.

Figure 6–4 Automatic Storage Management Create Disk Group Page



Create Disk Group

* Name

Redundancy ○ HIGH ● NORMAL ○ EXTERNAL

Member Disks

Select Member Disks: Only Candidate Disks

Select	Path	Header Status	Label	ASM Disk Name	Size	Size Unit	By Failure Group	Force Usage
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk1	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk2	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk3	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk4	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/test_disk4	CANDIDATE			300	MB		□

Disk Group Compatibility

Higher compatibility enables the user to use new features available in the newer version. Once the disk group is created, the compatibility version can not be lowered.

Database Compatibility: 10.1
The minimum software version required for a database instance to use files in this disk group.

ASM Compatibility: 10.1
The minimum software version required for an ASM instance to mount this disk group.

TIP The database compatibility has to be less than or equal to the ASM compatibility.

5. If you want to display not only disks that are available to be added to a disk group, but all ASM disks, including disks that already belong to a disk group (header status = `MEMBER`), then select **All Disks** from the Select Member Disks list.

   The page is re-displayed with the complete list of ASM disks that were discovered by the ASM instance.

6. Enter the disk name, select a redundancy type (high, normal, or external), and optionally, you can enter a failure group name.

7. Select disks as follows:

   - Select the box to the left of each disk that you want to include in the new disk group.
   - If you want to force the inclusion of a disk in the disk group, select the **Force Usage** box for that disk.

   Caution: The Force Usage box causes the disk to be added to the new disk group even if the disk already belongs to another disk group and has valid database data. This data will be lost. You must be certain that you are selecting a disk that can legitimately be added to the disk group. See "Creating Disk Groups" on page 4-2 for a discussion of the `FORCE` option.

8. Specify a **Disk Compatibility** value for the disk group to enable your system to use the latest ASM features.

   Note: You cannot reverse setting a disk compatibility value.

- For setting a disk group's database compatibility, the value that you use must be less than or equal to the ASM compatibility. This is the minimum software version that is required for a database instance to use the files in the disk group. If the database compatibility value that you use is greater than the ASM compatibility, then Enterprise Manager displays the following: `Error: The database compatibility has to be less or equal to the ASM compatibility.` If you enter a value for a lower version than the current version that is in use, then Enterprise Manager displays the error: `The version to advance to has to be greater than the current version.` By default, both boxes are checked.

> **Note:** You cannot modify the database compatibility for an ASM cluster that is in a rolling migration state.

- For setting a disk group's ASM Compatibility, the value that you use is the minimum software version required for an ASM instance to mount this disk group.

9. Click **OK** to create the disk group or click **Cancel** to stop the disk group creation operation. You can also click **Show SQL** to show the SQL statements that the disk group creation operation uses.

> **See Also:** See "Creating Disk Groups" on page 4-2 for more information about creating disk groups the *Oracle Database Reference* description of the `V$ASM_DISK` view for information about the various header statuses of ASM disks

Adding Disks to Disk Groups

You can add disks to a disk group to increase the total amount of storage space in a disk group. If you add multiple disks in a single operation, then ASM rebalances the disk group to evenly distribute the data on all disks, including the newly added disks.

You can control the power of a rebalance operation, which is a number from 0 to 11. The higher the number, the faster the rebalance operation completes. Lower numbers cause rebalancing to take longer, but consume fewer processing and I/O resources. This leaves these resources available for the database. The default value of `1` minimizes disruption to the database.

A value of `0` blocks the rebalance operation altogether. Manual or automatic rebalance can then occur at a later time. You might want to postpone rebalancing to wait for a time when there are fewer demands on the database, or to add more disks or drop disks later if you want the rebalancing to be done only once for all of your disk group changes.

> **See Also:** "Altering Disk Groups" on page 4-4 for information about controlling rebalance operations

To add one or more disks to a disk group:

1. Access the Automatic Storage Management Home page.

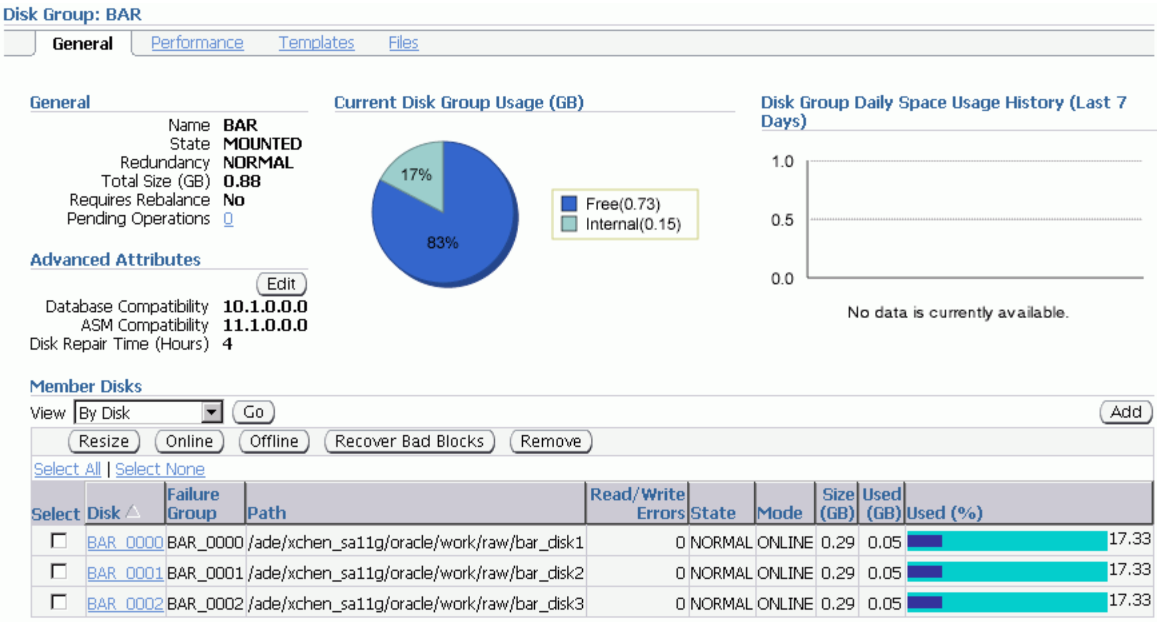
> **See Also:** "The Automatic Storage Management Home Page" on page 6-1 for instructions

2. Click the **Disk Groups** link to display the Disk Groups property page.

3. If the ASM Login page appears, then log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password for the ASM instance. This password was set up when the ASM instance was created.

4. Click a link in the **Name** column to select the disk group to which you want to add disks. The Disk Group page similar to the page in Figure 6–5 appears, displaying a list of disks that are already in the disk group.

Figure 6–5 Automatic Storage Management Disk Group Page



Disk Group: BAR

General

Name	BAR
State	MOUNTED
Redundancy	NORMAL
Total Size (GB)	0.88
Requires Rebalance	No
Pending Operations	0

Advanced Attributes

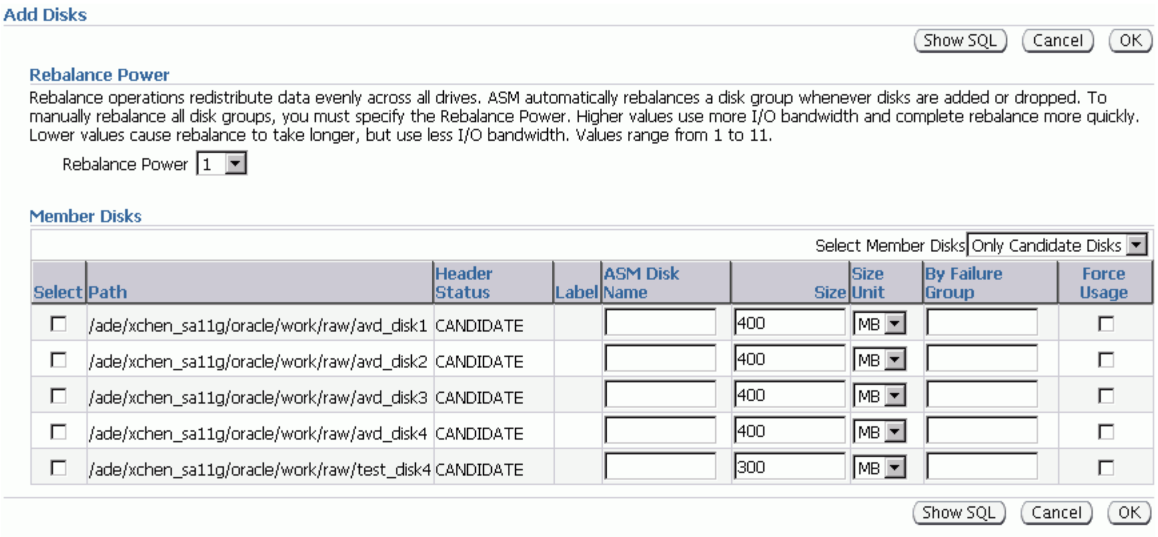
Database Compatibility	10.1.0.0.0
ASM Compatibility	11.1.0.0.0
Disk Repair Time (Hours)	4

Member Disks

Select	Disk	Failure Group	Path	Read/Write Errors	State	Mode	Size (GB)	Used (GB)	Used (%)
□	BAR_0000	BAR_0000	/ade/xchen_sa11g/oracle/work/raw/bar_disk1	0	NORMAL	ONLINE	0.29	0.05	17.33
□	BAR_0001	BAR_0001	/ade/xchen_sa11g/oracle/work/raw/bar_disk2	0	NORMAL	ONLINE	0.29	0.05	17.33
□	BAR_0002	BAR_0002	/ade/xchen_sa11g/oracle/work/raw/bar_disk3	0	NORMAL	ONLINE	0.29	0.05	17.33

5. Click **Add** and Enterprise Manager displays the Add Disks page. This page displays a list of ASM disks that are available to be added to the disk group. This includes disks with the header status of `CANDIDATE`, `PROVISIONED`, or `FORMER`, and so on.

Figure 6–6 Automatic Storage Management Add Disks Page



Add Disks

Rebalance Power

Rebalance operations redistribute data evenly across all drives. ASM automatically rebalances a disk group whenever disks are added or dropped. To manually rebalance all disk groups, you must specify the Rebalance Power. Higher values use more I/O bandwidth and complete rebalance more quickly. Lower values cause rebalance to take longer, but use less I/O bandwidth. Values range from 1 to 11.

Rebalance Power 1

Member Disks

Select Member Disks Only Candidate Disks

Select	Path	Header Status	Label	ASM Disk Name	Size	Size Unit	By Failure Group	Force Usage
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk1	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk2	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk3	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/avd_disk4	CANDIDATE			400	MB		□
□	/ade/xchen_sa11g/oracle/work/raw/test_disk4	CANDIDATE			300	MB		□

6. To display not only disks that are available to be added to a disk group, but all of the ASM disks, including disks that already belong to a disk group (header status = `MEMBER`), select **All Disks** from the Select Member Disks drop-down list on the right hand side of the page. Enterprise Manager re-displays the page with the complete list of ASM disks that were discovered by the ASM instance.

7. Optionally change the rebalance power by selecting a value from the Rebalance Power list.

8. Select disks as follows:

   - Select the box to the left of each disk that you want to add to the disk group.
   - If you want to force the inclusion of a disk in the disk group, select the **Force Reuse** box in the right-most column.

   > **Caution:** The Force Reuse box causes the disk to be added to the disk group even if the disk already belongs to another disk group and has valid database data. This data will be lost. You must be certain that you are selecting a disk that can legitimately be added to the disk group. Refer to "Adding Disks to a Disk Group" on page 4-5 for an example of the `FORCE` option.

   - Optionally enter an ASM disk name for each disk. ASM provides a name if you do not.

9. Click **OK** to add the selected disks.

   See Also: "Altering Disk Groups" on page 4-4 for more information about modifying disk group properties

Dropping Disks from Disk Groups

When you drop a disk from a disk group, ASM rebalances the disk group by moving all of the file extents from the dropped disk to other disks in the disk group. ASM then releases the disk and you can then add the disk to another disk group or use it for other purposes. You can drop one or more disks in a single operation. You can also optionally set rebalance power for the drop operation. The following is a list of possible reasons for dropping a disk:

- A disk is starting to fail and you need to replace it
- You want to upgrade a disk
- You want to reallocate the disk to a different disk group or reallocate it to a different storage system

> **Note:** Dropping disks from a disk group only logically deletes the disks from the disk group. It does not delete the disk contents. However, the contents are lost when you add the disk to a new disk group.

To drop one or more disks from a disk group:

1. Access the Automatic Storage Management Home page.

   See "The Automatic Storage Management Home Page" on page 6-1 for instructions.

2. Click the **Disk Groups** link to display the Disk Groups property page.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password for the ASM instance. This password was set up when the ASM instance was created.
4. Click a link in the **Name** column to select the disk group from which you want to drop disks.

   The Disk Group page appears.
5. Under the Member Disks heading, in the Select column, select the boxes for the disks that you want to drop, and then click **Remove**.

   A confirmation page appears.
6. If you want to change rebalance power, or if you want set the `FORCE` option for dropping disks, do the following:

   a. Click **Show Advanced Options**.

   b. Optionally select the **Force** option or select a rebalance power in the Rebalance Power list.

   Note: You might need the `FORCE` option if ASM cannot read from or write to the disk. See "Dropping Disks from Disk Groups" on page 4-6 for a discussion of the `FORCE` option.
7. Click **Yes** to drop the disks.

   The Disk Group page returns, and displays a state of `DROPPING` for each disk being dropped.
8. Refresh the page until the dropped disks no longer appear in the disk group.

   Caution: You cannot reuse or disconnect the dropped disks until the drop and rebalance operations are complete. Operations are complete when the dropped disks no longer appear in the disk group.See "Dropping Disks from Disk Groups" on page 4-6 for more information, and for other caveats for dropping disks.

Dropping Disk Groups

When you drop a disk group, you delete the disk group and optionally all of its files. You cannot drop a disk group if any of its database files are open. ASM deletes all of the diskgroup contents only if you specify the "including contents" option. After dropping a disk group, you can add its member disks to other disk groups or use them for other purposes.

One reason to drop a disk group is to change the redundancy level (normal, high, or external). Because you cannot change the redundancy of a disk group, you must drop the disk group and then re-create it with the proper redundancy. In this case, you must back up or move disk group data before you drop the disk group.

To drop a disk group:

1. Access the ASM Home page.

See Also: "The Automatic Storage Management Home Page" on page 6-1 for instructions about how to access the Automatic Storage Management Home page

2. Click the **Disk Group** tab to display the Disk Groups page.
3. If the ASM Login page appears, log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password for the ASM instance. This password was set up when the ASM instance was created.
4. Click the box in the **Select** column to select a disk group that you want to drop.
5. Click **Delete** and confirmation page appears.
6. If you want to delete the disk group even if it still contains files, click **Show Advanced Options** and ensure that the **Including Contents** option is selected. If the disk group contains files and the **Including Contents** option is not selected, then you cannot drop the disk group.
7. To delete a disk group that belongs to a disk group that you cannot mount, use the `FORCE` option in the Advanced Options confirmation dialog. The `FORCE` option applies to the `DISMOUNTED` disk group only. This option enables you to delete a disk header and mark it as "former".

Caution: Use extreme care when using the `FORCE` option because the ASM instance does not verify whether the disk group is used by any other ASM instance before ASM deletes the disk group.

8. On the confirmation page, click **Yes**.

See Also: "Dropping Disk Groups" on page 4-28 for more information about dropping disks

Monitoring Disk Group Usage

Perform the following procedures to monitor disk group usage:

To view ASM disk group usage and free space:

1. Access the Automatic Storage Management Home page.

See Also: "The Automatic Storage Management Home Page" on page 6-1 for instructions about how to access the home page

2. Click the **Disk Groups** link to view the Disk Groups property page.
3. If the ASM Login page appears, then log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set up for ASM when the ASM instance was created.

The Disk Group list page, similar to the page shown in Figure 6–7, displays all of the disk groups and their space usage information.

The Usable Free column displays the space in megabytes that is actually available in the disk group. The values in this column take into account the redundancy level of the disk group, and exclude the space that the disk group reserves for restoring full redundancy for all files in the disk group after a disk failure.

See Also: "Managing Capacity in Disk Groups" on page 4-11 and "Using Views to Obtain ASM Information" on page 4-29

Bringing Disks Online and Offline

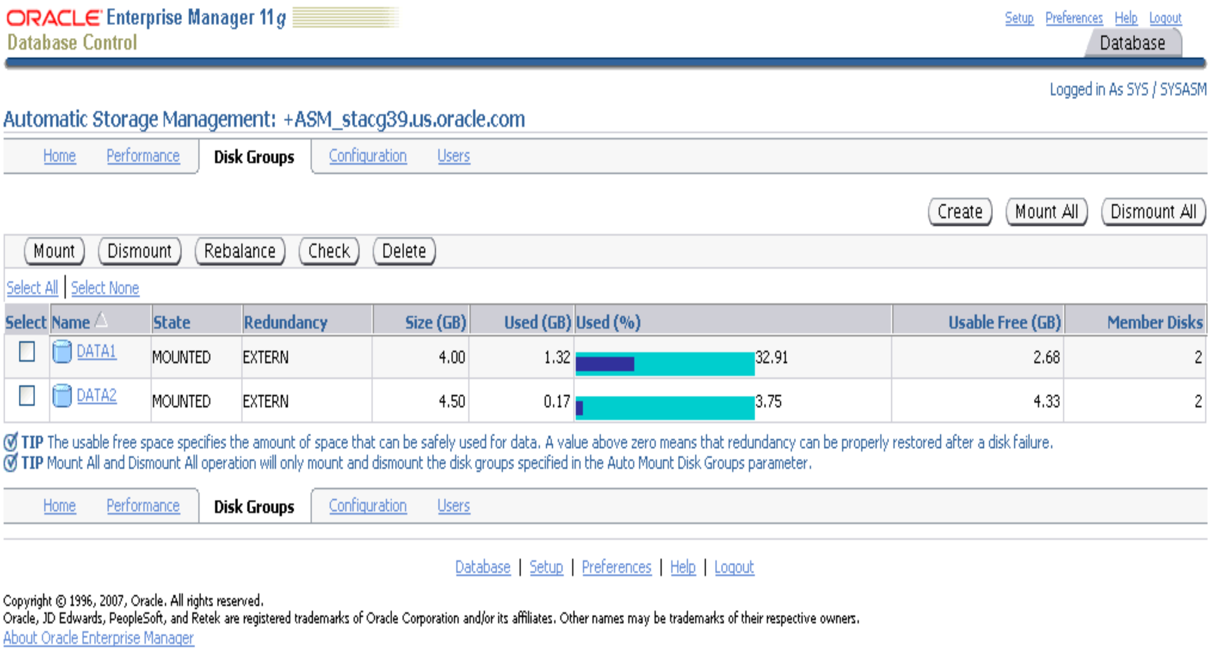
Perform the following procedures to bring a disk online or offline:

1. Access the Automatic Storage Management Home page.

   See "The Automatic Storage Management Home Page" on page 6-1 for instructions.

2. Click the **Disk Groups** link at the top of the page to view the Disk Groups page. Enterprise Manager displays a page similar to the one show in Figure 6–7.

Figure 6–7 Automatic Storage Management Disk Groups List Page


ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Database
Home Performance Disk Groups Configuration Users
Create Mount All Dismount All
Mount Dismount Rebalance Check Delete
Select All | Select None
Home Performance Disk Groups Configuration Users
Database | Setup | Preferences | Help | Logout

Logged in As SYS / SYSASM

Automatic Storage Management: +ASM_stacg39.us.oracle.com

Select	Name	State	Redundancy	Size (GB)	Used (GB)	Used (%)	Usable Free (GB)	Member Disks
□	DATA1	MOUNTED	EXTERN	4.00	1.32	32.91	2.68	2
□	DATA2	MOUNTED	EXTERN	4.50	0.17	3.75	4.33	2

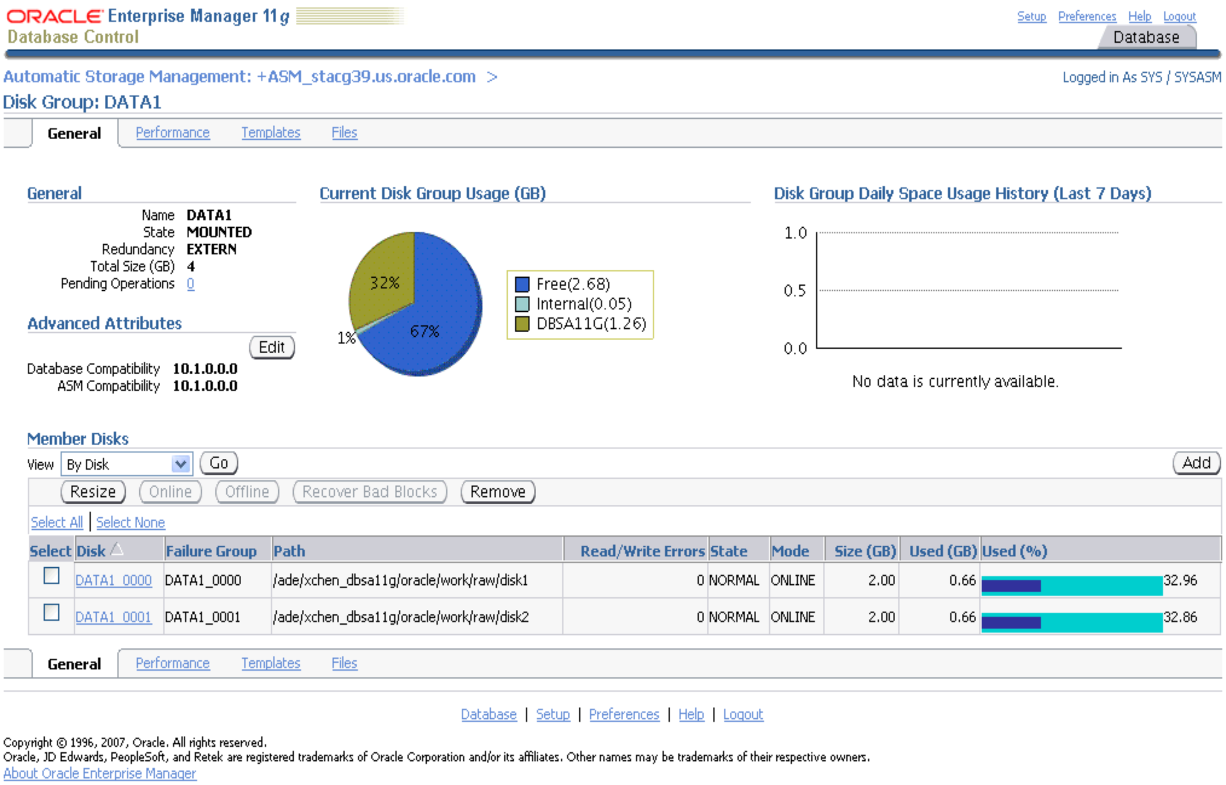
TIP The usable free space specifies the amount of space that can be safely used for data. A value above zero means that redundancy can be properly restored after a disk failure.

TIP Mount All and Dismount All operation will only mount and dismount the disk groups specified in the Auto Mount Disk Groups parameter.

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

3. Click the name of a disk in the Name column and Enterprise Manager displays the General page for the disk group that you want to modify. The General page that you see should be similar to the one shown in Figure 6–8.

Figure 6–8 Automatic Storage Management Disk Group Details Page



Automatic Storage Management: +ASM_stacg39.us.oracle.com >

Logged in As SYS / SYSASM

Disk Group: DATA1

General

Name	DATA1
State	MOUNTED
Redundancy	EXTERN
Total Size (GB)	4
Pending Operations	0

Advanced Attributes

Database Compatibility	10.1.0.0.0
ASM Compatibility	10.1.0.0.0

Current Disk Group Usage (GB)

- 🟦 Free(2.68)
- ▨ Internal(0.05)
- 🟩 DBSA11G(1.26)

Disk Group Daily Space Usage History (Last 7 Days)

No data is currently available.

Member Disks

Select	Disk	Failure Group	Path	Read/Write Errors	State	Mode	Size (GB)	Used (GB)	Used (%)
□	DATA1_0000	DATA1_0000	/ade/xchen_dbsa11g/oracle/work/raw/disk1	0	NORMAL	ONLINE	2.00	0.66	32.96
□	DATA1_0001	DATA1_0001	/ade/xchen_dbsa11g/oracle/work/raw/disk2	0	NORMAL	ONLINE	2.00	0.66	32.86

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

4. To bring a disk online, select the disk that you want to bring online and click **Online**. Note that if you bring a disk online, then ASM performs the following:

   - ASM performs the online operation on the selected disks. You can select one or more or disks, or all of the disks to bring online. If you select all of the disks, then Oracle uses the `ALL` keyword for this operation. Otherwise, only the selected disks are affected by the online operation.
   - During the online process, Enterprise Manager first places a disk into online mode to accommodate write operations. Then Enterprise Manager resynchronizes stale data on the disk with the current data until the disk is fully online for read operations; this is the ASM fast mirror resync feature.
   - You can also choose the By Failure Group view to perform this operation on a selected disk group. In this case, Oracle uses the SQL for `disks in failure group`.
   - You can also select a disk that is already online, but Enterprise Manager ignores that selection and only brings disks online that were previously offline.
   - Click **Yes** on the Enterprise Manager Confirmation dialog to begin bringing the disk online or click **No** to cancel the operation. You can also click **Show SQL** to review the SQL that Enterprise Manager uses for the online operation.

5. To take a disk offline, select the disk that you want to take offline and click **Offline**. Note that if you bring a disk offline, then ASM performs the following:

- Depending on how much data is stale, the operation may require additional time to complete. You can monitor the operation's progress using the Pending Operations link from the Disk Group Home page.
- The value that you set for Rebalance Power using ASM_POWER_LIMIT must be a whole number ranging from 0 to 11, inclusively. Note the text on the page that describes how the value of Rebalance Power affects the fast mirror resync operation. If the ASM_POWER_LIMIT is set to 0, then ASM does not perform automatic rebalance.
- By default, the Rebalance Power value uses the value of the ASM_POWER_LIMIT initialization parameter.
- Enterprise Manager displays an Offline Confirmation page and the text on this page explains that after Enterprise Manager takes the selected disks offline, the ASM instance waits for the period of time that is specified by the value that you have set for the DISK_REPAIR_TIME initialization parameter before ASM drops the disks. During the disk repair time period, you can repair the disks and then place them back online. For each disk that you successfully place online, the pending drop operation is canceled and a resynchronization operation begins.
- If you have set DISK_REPAIR_TIME to a large value, for example, greater than one day, then this increases the possibility of data loss. This is because the files in the disk group are protected with a lower degree of redundancy during this period.

  Click **Yes** on the Enterprise Manager Conformation dialog to begin the offline processing or click **No** to stop the operation. You can also click **Show SQL** to see the SQL that Enterprise Manager uses for the offline operation.

Mounting and Dismounting Disk Groups

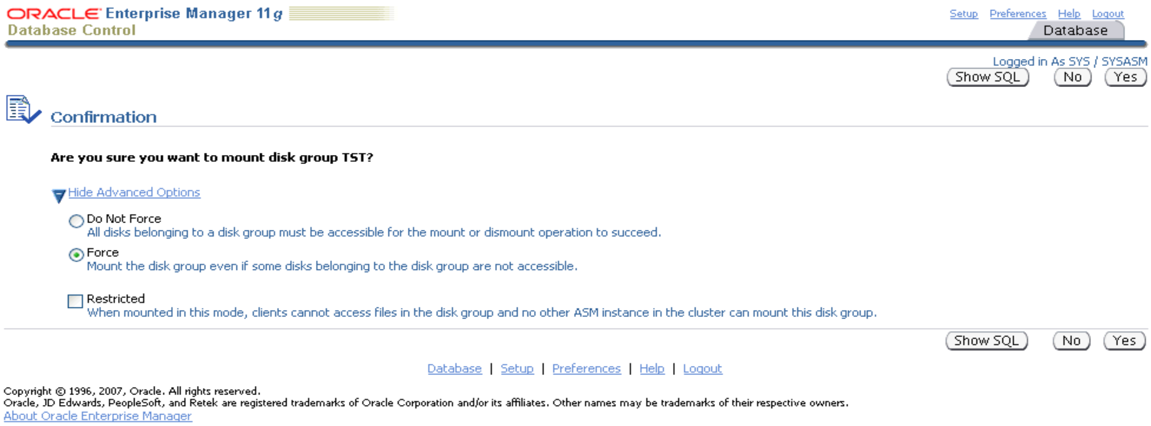
Use the FORCE option to mount a disk group when there are missing disks or missing failure groups. This enables you to correct configuration errors, such as incorrect values for ASM_DISKSTRING without incurring unnecessary rebalance operations.

Disk groups mounted with the FORCE option have one or more disks offline if the disks were not available at the time of the mount. You must take corrective action to restore those devices before the time set with the DISK_REPAIR_TIME value expires. Failing to restore and put those disks back online within the disk repair time frame results in ASM automatically removing the disks from the disk group. This would then require a rebalance operation to restore redundancy for all of the files in the disk group. Perform the following procedures to mount a disk with the FORCE option:

1. From the ASM home page click the **Disk Groups** tab.
2. Select the disk group that has a disk that you need to repair and click **Dismount**. Click **OK** on the Dismount confirmation dialog. The State column of the Disk Group home page should show the Status as DISMOUNTED.
3. Perform any needed repairs to the disk in the disk group.
4. Click the box in the Select column next to the disk that you want to force mount and click **Mount**.
5. On the Disk Mount Confirmation dialog, click the **Show Advanced Options** link.
6. On the Mount Force Confirmation dialog as shown inFigure 6–9, select the **Force** option and click **Yes**. You can also click **No** to cancel the mount force operation

and **Show SQL** to review the SQL that Enterprise Manager uses to perform the mount force operation.

Figure 6–9 Mount Force Confirmation Dialog


ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Database
Logged in As SYS / SYSASM
Show SQL No Yes
Show SQL No Yes
Database | Setup | Preferences | Help | Logout
About Oracle Enterprise Manager

Confirmation

Are you sure you want to mount disk group TST?

Hide Advanced Options

○ Do Not Force
All disks belonging to a disk group must be accessible for the mount or dismount operation to succeed.

● Force
Mount the disk group even if some disks belonging to the disk group are not accessible.

□ Restricted
When mounted in this mode, clients cannot access files in the disk group and no other ASM instance in the cluster can mount this disk group.

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Administering Advanced Disk Group Properties

This section explains how to administer advanced disk group properties under the following topic:

- Configuring Disk Repair Time

Configuring Disk Repair Time

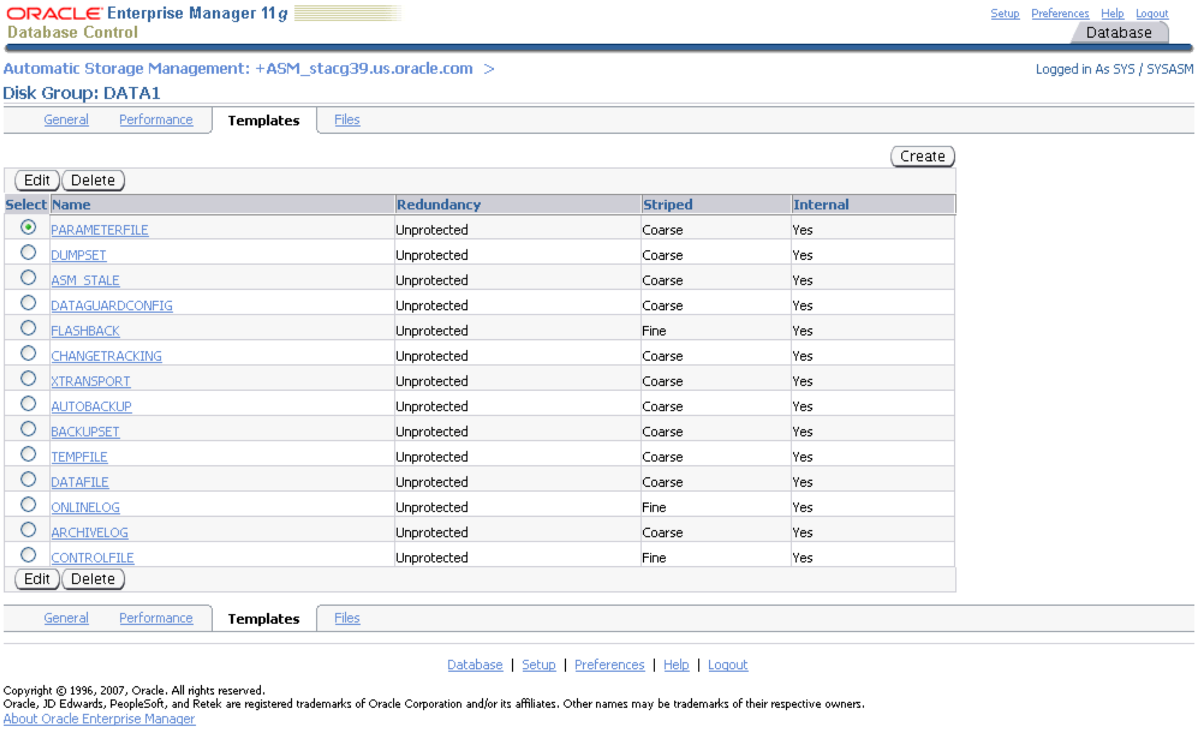
Perform the following steps to change the value for the `ASM_DISK_REPAIR_TIME` parameter:

1. On the Automatic Storage Disk Group List page, click a disk group name and Enterprise Manager displays the Disk Group Home page.
2. Click **Edit** under the Advanced Attributes section, enter a disk repair time value, and click **Apply**.

Managing Disk Group Templates

You can manage disk group templates with the Enterprise Manager Templates page, shown in Figure 6–10.

Figure 6–10 Automatic Storage Management Templates Page



ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Database

Automatic Storage Management: +ASM_stacg39.us.oracle.com >

Logged in As SYS / SYSASM

Disk Group: DATA1

General Performance Templates Files
Create
Edit Delete

Select	Name	Redundancy	Striped	Internal
●	PARAMETERFILE	Unprotected	Coarse	Yes
○	DUMPSET	Unprotected	Coarse	Yes
○	ASM_STALE	Unprotected	Coarse	Yes
○	DATAGUARDCONFIG	Unprotected	Coarse	Yes
○	FLASHBACK	Unprotected	Fine	Yes
○	CHANGETRACKING	Unprotected	Coarse	Yes
○	XTRANSPORT	Unprotected	Coarse	Yes
○	AUTOBACKUP	Unprotected	Coarse	Yes
○	BACKUPSET	Unprotected	Coarse	Yes
○	TEMPFILE	Unprotected	Coarse	Yes
○	DATAFILE	Unprotected	Coarse	Yes
○	ONLINELOG	Unprotected	Fine	Yes
○	ARCHIVELOG	Unprotected	Coarse	Yes
○	CONTROLFILE	Unprotected	Fine	Yes

Edit Delete
General Performance Templates Files
Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

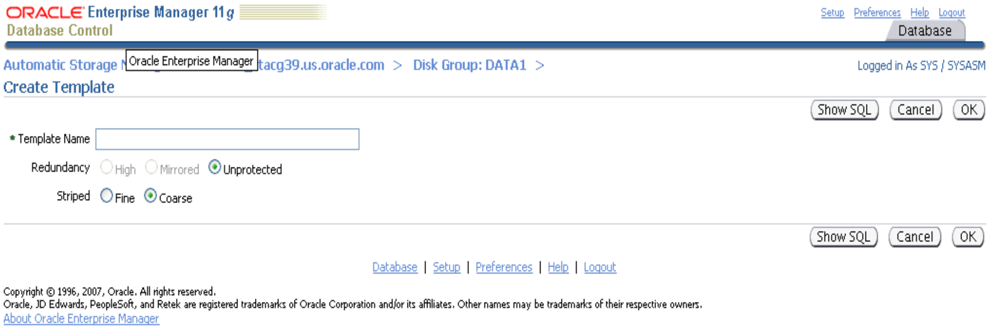
To manage ASM templates:

1. Access the Automatic Storage Management Home page.

   See "The Automatic Storage Management Home Page" on page 6-1 for instructions.

2. Click the **Disk Groups** link to display the Disk Groups property page.
3. Click a link in the **Name** column to select a disk group.
4. Click the **Templates** link to display the Templates page.
5. You can click **Create** to add a new template, click **Edit** to change a page, or click **Delete** to remove a template.

   If you click **Create**, then the Create Template page displays and you can enter a name in the Template Name field. You can also specify a Redundancy level and Striped granularity.



ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Database

Automatic Storage M Oracle Enterprise Manager tacg39.us.oracle.com > Disk Group: DATA1 >

Logged in As SYS / SYSASM

Create Template

* Template Name

Redundancy ○ High ○ Mirrored ● Unprotected

Striped ○ Fine ● Coarse

Show SQL Cancel OK
Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

See "Managing Disk Group Templates" on page 5-15.

Monitoring ASM Performance

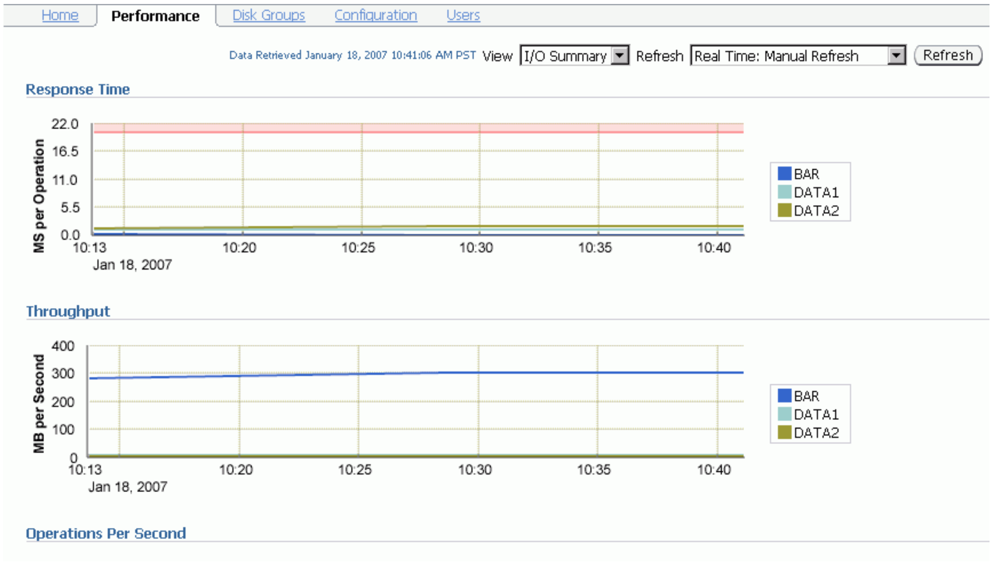
This section describes how to perform the following performance-related ASM administrative tasks:

- Monitoring General ASM Performance
- Checking Disk Group Integrity

Monitoring General ASM Performance

You can monitor ASM performance with the Enterprise Manager Performance page, shown in Figure 6–11.

Figure 6–11 Automatic Storage Management Performance Page



Data Retrieved January 18, 2007 10:41:06 AM PST

View: I/O Summary

Refresh: Real Time: Manual Refresh

To monitor ASM performance:

1. Access the Automatic Storage Management Home page.

   See Also: "The Automatic Storage Management Home Page" on page 6-1 for more information about accessing the home page

2. Click the **Performance** link to view the Performance property page.
3. Determine the view and refresh rate of the performance graphs with the **View** and **Refresh** options. Click **Refresh** to manually refresh the graphs. The refresh option has the following two intervals:
   - Manual refresh: manual, 15 second interval, 30 second interval
   - Historical data: Last 24 hours, last 7 days, last 31 days

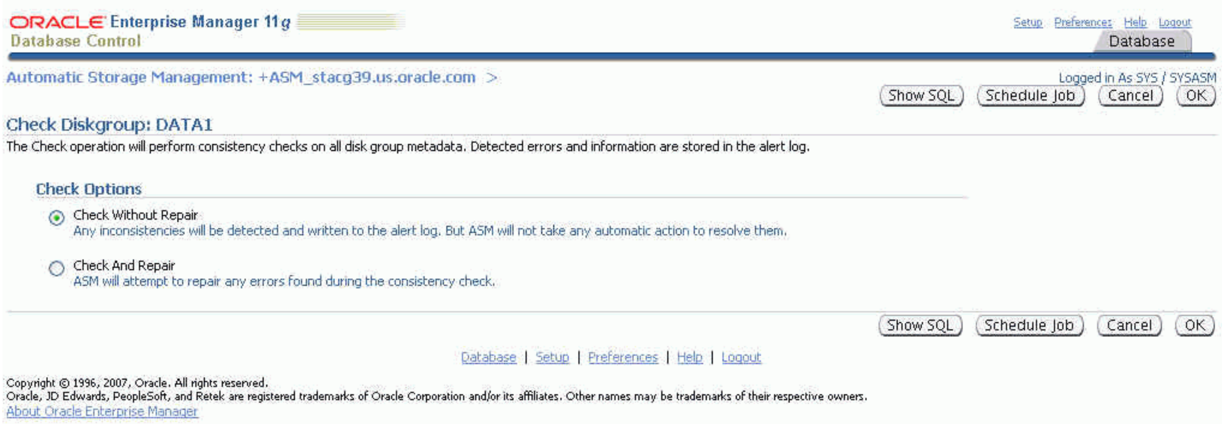
   See Also: "Performance and Scalability Considerations for Disk Groups" on page 4-20

Checking Disk Group Integrity

When you check the integrity of ASM disk groups, Enterprise Manager performs consistency checks on all of the disk group metadata. If ASM detects errors, then ASM stores information about errors in the alert log. To check disk groups, perform the following steps:

1. From the Disk Groups Administration page click **Check** and ASM displays a confirmation dialog for the check disk group operation.

Figure 6–12 Automatic Storage Management Check Diskgroup Confirmation Dialog



Automatic Storage Management: +ASM_stacg39.us.oracle.com >

Logged in As SYS / SYSASM

Check Diskgroup: DATA1

The Check operation will perform consistency checks on all disk group metadata. Detected errors and information are stored in the alert log.

Check Options

● Check Without Repair
Any inconsistencies will be detected and written to the alert log. But ASM will not take any automatic action to resolve them.

○ Check And Repair
ASM will attempt to repair any errors found during the consistency check.

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

2. Select **Check Without Repair** if you only want ASM to record information about disk group inconsistencies in the alert log. In this operation, ASM will not resolve the errors. Select **Check And Repair** if you want ASM to attempt to repair errors that Enterprise Manager identifies during the disk group consistency check.

3. Click **Yes** to start the disk group check operation or **No** to cancel it. You can also select **Show SQL** to review the SQL statements that the disk group check operation uses. Click **OK** to run the check operation as a command.

4. To schedule a job, click **Schedule Job**.

5. Optionally, you can modify or re-enter the job name and description. Also provide host credentials and enter the scheduling options by selecting **Immediately** or **Later** and modifying the date and time as needed.

6. Click **OK** to schedule the job or **Cancel** to terminate the job creation process.

Backing Up ASM Files

Oracle recommends that you use RMAN to back up files managed by ASM. The procedures for using RMAN are documented in *Oracle Database 2 Day DBA*.

You can also back up your ASM-managed files by performing a files related operation using the following procedures:

1. Create an alias.
2. Create a directory
3. Delete the alias and directory.
4. Rename the alias and directory

> **See Also:** For additional information about RMAN, refer to the *Oracle Database Backup and Recovery User's Guide*.

Performing Bad Block Recovery

The procedures that you use in Enterprise Manager to perform bad block recovery correspond to the steps that you could manually perform with the `remap` command using the ASM Command utility (ASMCMD). Use the following procedures to perform bad block recovery:

1. Go to the Automatic Storage Management home page.
2. Click the **Disk Groups** link at the top of the page.
3. If you are not already logged in, then the Log In page appears and you should log in as the SYS user, connecting as SYSASM. Provide the SYS password for the ASM instance. This password was set up when the ASM instance was created.
4. Click one of the disk group names and Enterprise Manager displays the disk group home page for that disk group.
5. Click a box in the Select column to the left of a disk for which you want to recover bad blocks and click **Recover Bad Blocks**.

> **Note:** You can only recover bad blocks for disk groups that you have configured with normal or high redundancy; you cannot use this feature for disk groups that you have configured with external redundancy.

Migrating to ASM

This section explains how to use Enterprise Manager to migrate your database storage to use ASM. Perform the following steps to migrate to ASM:

1. At the top of the Enterprise Manager Database Instance Home Page, click the link labeled **Server**. Enterprise Manager displays Database Server Page.
2. On the top left-hand side of the Database Server Page under the Storage heading, click the link labeled **Migrate to ASM**.
3. If prompted for ASM login credentials, then enter the user `SYS`, provide the `SYS` password that was set for the ASM instance upon installation, and connect as `SYSDBA`. Enterprise Manager displays the Migrate Database To ASM: Migration Options Page.

   See Also: "Authentication for Accessing ASM Instances" on page 3-17 for more information about authentication

4. On the Migrate Database To ASM: Migration Options Page, which is the first step in a four-step process, verify the information about the files that you want to migrate to ASM. If the target database is in `ARCHIVELOG` mode, then you could perform online or offline migration by selecting the corresponding radio button. Otherwise, using offline migration is the default. If you perform an online migration, then you can suspend the migration by selecting the checkbox next to Suspend Migration Job Before Switching Database To ASM. Additionally, you can enter an email address to which Enterprise Manager sends a message to notify you about the suspension. By default on this page, the Recovery-related Files option is selected. Deselect this option if you do not want to use it. Also verify the entry for the initialization parameter file. Lastly, the **Parallelize File Copy Operations** option is selected by default. Deselect this if you do not want Enterprise Manager to perform copy operations in parallel. Click **Next** when you complete your

selections on this page and Enterprise Manager displays the **Migrate Database To ASM**: **Disk Group Options** page.

5. On the Disk Group Options page under Database Area, verify the location where ASM should create datafiles, temporary files, redo log files, and control files. Also verify the Flash Recovery Area where ASM should create the recovery-related files such as archived redo log files, backup files, and so on. Also under Flash Recovery Area, the option for Setup Flash Recovery Area without Migrating Recovery-related Files is selected by default. If needed, change the selection to migrate the files. Click **Next** when you complete your selections on the Disk Group Options page.
6. If you chose to use a Flash Recovery Area and Enterprise Manager displays a warning that the available space on the disk group is less than the specified Flash Recovery Area size for the migration, then you can either continue by clicking **Yes**, or click **No** to stop the operation and modify your storage.
7. When you continue, Enterprise Manager displays the Schedule page on which you can specify a job name and a date and time at which you want the migration job to begin. Click **Next** when you have configured your desired settings.
8. Enterprise Manager displays the Review page on which you can review all of your selections before beginning your migration operation. To expand the tablespace details and review them, click the plus (+) sign next to Tablespaces and Files To Be Migrated. Click **Back** to modify any of your selections or **Submit Job** to begin the migration. Click **Cancel** to cancel the operation.

Caution: If your migration job fails, then do not start your database until you complete a recovery operation. Otherwise, you may corrupt your database.

9. After you submit your job, Enterprise Manager should display a confirmation dialog indicating that the migration job was successful. Click **View Status** on the confirmation dialog to examine the progress of your migration job.

ASM: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through ASM tasks and includes annotated screen shots.

To view the ASM OBE, point your browser to the following location:

`http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/2day_dba/asm/asm.htm`

7

ASM Command-Line Utility

This chapter describes the Automatic Storage Management (ASM) Command-Line Utility (ASMCMD). This chapter contains the following topics:

- About ASMCMD
- About ASM Files, Filenames, Directories, and Aliases
- Running ASMCMD
- ASMCMD Command Reference

About ASMCMD

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.

Note: You cannot use ASMCMD to create or drop disk groups or to add or drop disks in a disk group. Use SQL commands for these operations.

About ASM Files, Filenames, Directories, and Aliases

ASMCMD works with ASM files, directories, and aliases. Before using ASMCMD, you should understand how these common computing concepts apply to the ASM environment.

This section contains the following topics about some key definitions:

- System-Generated Filename or Fully Qualified Filename
- Directory
- Alias
- Absolute Path and Relative Path
- Wildcard Characters

System-Generated Filename or Fully Qualified Filename

Every file created in ASM gets a system-generated filename, otherwise known as a fully qualified filename. This is the same as a complete path name in a local file system.

ASM generates filenames according to the following scheme:

```
+diskGroupName/databaseName/fileType/fileTypeTag.fileNumber.incarnation
```

An example of a fully qualified filename is the following:

```
+dgroup2/sample/CONTROLFILE/Current.256.541956473
```

In the previous fully qualified filename, `dgroup2` is the disk group name, `sample` is the database name, `CONTROLFILE` is the file type, and so on.

Only the forward slash (/) is supported by ASMCMD. Filenames are not case sensitive, but are case retentive. If you type a path name as lowercase, ASMCMD retains the lowercase.

For more information about ASM filenames, refer to "Fully Qualified ASM Filename" on page 5-4.

Directory

As in other file systems, an ASM directory is a container for files, and an ASM directory can be part of a tree structure of other directories. The fully qualified filename represents a hierarchy of directories in which the plus sign (+) represent the root directory. In each disk group, ASM automatically creates a directory hierarchy that corresponds to the structure of the fully qualified filenames in the disk group. The directories in this hierarchy are known as system-generated directories.

ASMCMD enables you to move up and down in this directory hierarchy with the `cd` (change directory) command. The ASMCMD `ls` (list directory) command lists the contents of the current directory, while the `pwd` command prints the name of the current directory.

When you start ASMCMD, the current directory is set to root (+). For an ASM instance with two disk groups, for example, `dgroup1` and `dgroup2`, entering an `ls` command with the root directory as the current directory produces the following output:

```
ASMCMD> ls
dgroup1/
dgroup2/
```

The following example demonstrates navigating the ASM directory tree (refer to the fully qualified filename shown previously):

```
ASMCMD> cd +dgroup1/sample/CONTROLFILE
ASMCMD> ls
Current.256.541956473
Current.257.541956475
```

You can also create your own directories as subdirectories of the system-generated directories using the `ALTER DISKGROUP` command or with the ASMCMD `mkdir` command. The directories that you create can have subdirectories, and you can navigate the hierarchy of both system-generated directories and user-created directories with the `cd` command.

The following example creates the directory `mydir` under `sample` in the disk group `dgroup1`:

```
ASMCMD> mkdir +dgroup1/sample/mydir
```

Note: The directory sample is a system-generated directory. The contents of dgroup1 represent the contents of disk group dgroup1.

If you start ASMCMD with the -p flag, then ASMCMD shows the current directory as part of its prompt. See "Including the Current Directory in the ASMCMD Prompt" on page 7-7.

```
ASMCMD [+] > cd dgroup1/mydir
ASMCMD [+dgroup1/mydir] >
```

Note that ASMCMD retains the case of the directory that you entered.

Alias

Aliases are filenames that are references or pointers to system-generated filenames. However, aliases are user-friendly names. Aliases are similar to symbolic links in UNIX or Linux computers. You can create aliases to simplify ASM filename administration. You can create aliases with an ALTER DISKGROUP command or with the mkalias ASMCMD command.

An alias has at a minimum the disk group name as part of its complete path. You can create aliases at the disk group level or in any system-generated or user-created subdirectory. The following are examples of aliases:

```
+dgroup1/ctl1.f
+dgroup1/sample/ctl1.f
+dgroup1/mydir/ctl1.f
```

If you run the ASMCMD ls (list directory) with the -l flag, each alias is listed with the system-generated file to which the alias refers.

```
ctl1.f => +dgroup2/sample/CONTROLFILE/Current.256.541956473
```

For more information about aliases, refer to "Alias ASM Filenames" on page 5-5.

Absolute Path and Relative Path

When you run an ASMCMD command that accepts a filename or directory name as an argument, you can use the name as either an absolute path or a relative path.

An absolute path refers to the full path of a file or directory. An absolute path begins with a plus sign (+) followed by a disk group name, followed by subsequent directories in the directory tree. The absolute path includes directories until the file or directory of interest is reached. A complete system-generated filename, otherwise known as the fully qualified filename, is an example of an absolute path to a file.

Using an absolute path enables the command to access the file or directory regardless of where the current directory is set. The following rm command uses an absolute path for the filename:

```
ASMCMD [+] > rm +dgroup2/sample/datafile/users.259.555341963
```

The following cd command uses an absolute path to the directory.

```
ASMCMD [+dgroup1/mydir] > cd +dgroup2/sample/CONTROLFILE
```

A relative path includes only the part of the filename or directory name that is not part of the current directory. That is, the path to the file or directory is relative to the current directory.

In the following example, the rm command operates on the file undotbs1.272.557429239, which in this case is a relative path. ASMCMD appends the current directory to the command argument to obtain the absolute path to the file. In this example this is +dgroup1/sample/DATAFILE/undotbs1.272.557429239.

```
ASMCMD [+] > cd +dgroup1
ASMCMD [+dgroup1] > cd sample/DATAFILE
ASMCMD [+dgroup1/sample/DATAFILE] > ls
EXAMPLE.269.555342243
SYSAUX.257.555341961
SYSTEM.256.555341961
UNDOTBS1.258.555341963
UNDOTBS1.272.557429239
USERS.259.555341963
ASMCMD [+dgroup1/sample/DATAFILE] > rm undotbs1.272.557429239
```

Paths to directories can also be relative. You can go up or down the hierarchy of the current directory tree branch by providing a directory argument to the cd command whose path is relative to the current directory.

In addition, you can use the pseudo-directories "." and ".." in place of a directory name. The "." pseudo-directory is the current directory. The ".." pseudo-directory is the parent directory of the current directory.

The following example demonstrates how to use relative directory paths and pseudo-directories:

```
ASMCMD [+dgroup1/sample] > cd DATAFILE
ASMCMD [+dgroup1/sample/DATAFILE] >cd ..
ASMCMD [+dgroup1/sample] >
```

Wildcard Characters

The wildcard characters * and % match zero or more characters anywhere within an absolute or relative path, which saves typing of the full directory or file name. The two wildcard characters behave identically. The ASMCMD commands that accept wildcards are cd, du, find, ls, lsdsk, and rm.

If a wildcard pattern matches only one directory when using wildcard characters with cd, then cd changes the directory to that destination. If the wildcard pattern matches multiple directories, then ASMCMD does not change the directory but instead returns an error.

The following examples illustrate the use of wildcards.

```
ASMCMD [+]> cd +dgroup1/sample/*FILE
asmcmd: *FILE: ambiguous

ASMCMD [+]> cd +dgroup1/sample/C*
ASMCMD [+dgroup1/sample/CONTROLFILE/]>

ASMCMD> ls +dgroup1/mydir1
ctl.f
data1.f
dummy.f

ASMCMD> ls +dgroup1/mydir1/d*
data1.f
dummy.f
```

```
ASMCMD> ls +group1/sample/*
+dgroup1/sample/CONTROLFILE/:
Current.260.555342185
Current.261.555342183

+dgroup1/sample/DATAFILE/:
EXAMPLE.269.555342243
SYSAUX.257.555341961
SYSTEM.256.555341961
UNDOTBS1.272.557429239
USERS.259.555341963

+dgroup1/sample/ONLINELOG/:
group_1.262.555342191
group_1.263.555342195
group_2.264.555342197
group_2.265.555342201

+dgroup1/sample/PARAMETERFILE/:
spfile.270.555342443

+dgroup1/sample/TEMPFILE/:
TEMP.268.555342229
```

Running ASMCMD

You can run the ASMCMD utility in either interactive or noninteractive mode. Before running ASMCMD, you must ensure that you are properly logged in and that your environment is properly configured.

This section contains the following topics:

- Preparing to Run ASMCMD
- Running ASMCMD in Interactive Mode
- Running ASMCMD in Noninteractive Mode
- Getting Help

Preparing to Run ASMCMD

Before running ASMCMD, check the following:

- To use most of the ASMCMD commands, ensure that the ASM instance is started and the ASM disk groups are mounted.

  If the ASM instance is not running or if the `ORACLE_SID` is set incorrectly, ASMCMD will still run, but only those commands that do not require an ASM instance will run. The commands are `lsdsk`, `help`, and `exit`. If you attempt to run other ASMCMD commands, an error message displays.

  Note: You cannot use ASMCMD to mount disk groups.

- Log in to the host on which the ASM instance that you want to work with is running. You must log in as a user that has SYSASM or SYSDBA privileges through operating system authentication. The SYSASM privilege is the preferred connection. See "Authentication for Accessing ASM Instances" on page 3-17.

  See Also: Refer to the *Oracle Database Administrator's Guide* for information about operating system authentication

- Set the ORACLE_HOME and ORACLE_SID environment variables to refer to the ASM instance. Depending on your operating system, you might have to set other environment variables to properly connect to the ASM instance.

  See Also: Refer to the *Oracle Database Administrator's Guide* for more information about setting environment variables

  The default value of the ASM SID for a single-instance database is +ASM. In Real Application Clusters environments, the default value of the ASM SID on any node is +ASM*node#*.

- Ensure that the bin subdirectory of your Oracle home is in your PATH environment variable.

Running ASMCMD in Interactive Mode

The interactive mode of the ASMCMD utility provides a shell-like environment where you are prompted to enter ASMCMD commands.

To run ASMCMD in interactive mode:

1. Enter the following at the operating system command prompt:

   ```
   asmcmd
   ```

   Oracle displays an ASMCMD command prompt as follows:

   ```
   ASMCMD>
   ```

2. Enter an ASMCMD command and press Enter. The command runs and displays its output, if any, and then ASMCMD prompts for the next command.
3. Continue entering ASMCMD commands. Enter the command exit to exit ASMCMD.

Displaying the ASMCMD Version Number

You can specify the -v option when starting asmcmd to displays the asmcmd version number. After displaying the version number, asmcmd immediately exits.

For example:

```
$ asmcmd -v
asmcmd version 11.1.0.3.0
```

Specifying the Type of Connection

You can specify the -a option to choose the type of connection, either SYSASM or SYSDBA. The default value is SYSASM.

For example:

```
$ asmcmd -a sysasm
```

Including the Current Directory in the ASMCMD Prompt

You can specify the `-p` option with the `asmcmd` command to include the current directory in the ASMCMD prompt as shown in the following example:

```
$ asmcmd -p
ASMCMD [+] > cd dgroup1
ASMCMD [+dgroup1] >
```

Running ASMCMD in Noninteractive Mode

In noninteractive mode, you run a single ASMCMD command by including the command and command arguments on the command line that invokes ASMCMD. ASMCMD runs the command, generates output if any, and then exits. The noninteractive mode is especially useful for running scripts.

To run ASMCMD in noninteractive mode where *`command`* is any valid ASMCMD command and *`arguments`* is a list of command flags and arguments, at the command prompt enter the following:

```
asmcmd command arguments
```

The following is an example of how to run ASMCMD in the noninteractive mode:

```
asmcmd ls -l
State    Type    Rebal  Unbal  Name
MOUNTED  NORMAL  N      N      DGROUP1/
MOUNTED  NORMAL  N      N      DGROUP2/
```

Getting Help

Type `help` at the ASMCMD prompt or as a command in noninteractive mode to view general ASMCMD help and a list of available commands.

You can type `help` *`command`* to display help text for a specific command. Also, you can type *`command`* `help` to display usage information about how to run the command with its options.

ASMCMD Command Reference

This section describes ASMCMD commands in detail. Table 7–1 provides a summary of all ASMCMD commands.

Table 7–1 Summary of ASM Commands

Command	Description
cd Command	Changes the current directory to the specified directory.
cp Command	Enables you to copy files between ASM disk groups on a local instan and remote instances.
du Command	Displays the total disk space occupied by ASM files in the specified ASM directory and all of its subdirectories, recursively.
exit Command	Exits ASMCMD.
find Command	Lists the paths of all occurrences of the specified name (with wildcarc under the specified directory.
help Command	Displays the syntax and description of ASMCMD commands.

Table 7–1 (Cont.) Summary of ASM Commands

Command	Description
ls Command	Lists the contents of an ASM directory, the attributes of the specified file, or the names and attributes of all disk groups.
lsct Command	Lists information about current ASM clients.
lsdg Command	Lists all disk groups and their attributes.
lsdsk Command	Lists disks visible to ASM.
md_backup Command	Creates a backup of all of the mounted disk groups.
md_restore Command	Restores disk groups from a backup.
mkalias Command	Creates an alias for system-generated filenames.
mkdir Command	Creates ASM directories.
pwd Command	Displays the path of the current ASM directory.
remap Command	Repairs a range of physical blocks on a disk.
rm Command	Deletes the specified ASM files or directories.
rmalias Command	Deletes the specified alias, retaining the file that the alias points to.

cd Command

Purpose

Changes the current directory to the specified directory.

Syntax and Description

`cd` *`dir_name`*

`dir_name` may be specified as either an absolute path or a relative path, including the `.` and `..` pseudo-directories. *`dir_name`* can contain wildcard characters. See "Wildcard Characters" on page 7-4.

Examples

```
ASMCMD [+dgroup2/hr] > cd +dgroup1/sample
ASMCMD [+dgroup1/sample] > cd DATAFILE
ASMCMD [+dgroup1/sample/DATAFILE] >cd ..
ASMCMD [+dgroup1/sample] >
```

cp Command

Purpose

Enables you to copy files between ASM disk groups on local instances to and from remote instances. The file copy cannot be between remote instances. The local ASM instance must be either the source or the target of the operation. You can also use this command to copy files from ASM disk groups to the operating system.

Syntax and Description

```
cp [-ifr] [connect_string:]src_fname [connect_string:]tgt_fname
cp [-ifr] [connect_string:]src_fnameN, src_fnameN+1...
   [connect_string:]tgt_directory
```

The syntax variables for the `cp` command are as follows:

- *connect_string* - The ASMCMD connection string for use with a remote instance copy.

  The *connect_string* parameter is not required for a local instance copy, which is the default case. In the case of a remote instance copy, you need to specify the connect string and ASM prompts for a password in a non-echoing prompt. The *connect_string* is in the form of:

  user_name@*host_name*[.*port_number*].*SID*

  The *user_name*, *host_name*, and *SID* are required. The default port number is `1521`.

- *src_fname*(s) - Source file name to copy from.

  Enter either the fully qualified file name, the system-generated name, or the ASM alias.

- *tgt_fname* - A user alias for the created target file name or alias directory name.
- *tgt_directory* - A target alias directory within an ASM disk group.

  The target directory must exist, otherwise the file copy returns an error.

The format of copied files is portable between Little-Endian and Big-Endian systems if the files exist in an ASM disk group. ASM automatically converts the format when it writes the files. For copying a non-ASM files from or to an ASM disk group, you can copy the file to a different endian platform and then use one of the commonly used utilities to convert the file.

Table 7–2 Flags for the cp Command

Flag	Description
-i	Interactive, prompt before copy file or overwrite
-f	Force, if an existing destination file, remove it and try again without user interaction
-r	Recursive, copy forwarding sub-directories recursively
-help	Displays help text.

Examples

```
ASMCMD[+]>cp +DG1/vdb.ctf1 /backups/vdb.ctf1
copying file(s)...
source +DG1/vdb.ctf1
target /backups/vdb.ctf1
file, /backups/vdb.ctf1, copy committed.

ASMCMD[+DG1]> cp vdb.ctf1 /tmp
copying file(s)...
source +DG1/vdb.ctf1
target /tmp/vdb.ctf1
file, /tmp/vdb.ctf1, copy committed.
```

du Command

Purpose

Displays the total space used for files in the specified directory and in the entire directory tree under the directory.

Syntax and Description

```
du [-H] [dir_name]
```

This command is similar to the `du -s` command on UNIX or Linux computers. If you do not specify *`dir_name`*, then information about the current directory is displayed. *`dir_name`* can contain wildcard characters. See "Wildcard Characters" on page 7-4.

The following two values are displayed, both in units of MB.

- `Used_MB` - This value does not include mirroring.
- `Mirror_used_MB` - This value includes mirroring.

For example, if a normal redundancy disk group contains 100 MB of data, then assuming that each file in the disk group is 2-way mirrored, `Used_MB` is 100 MB and `Mirror_used_MB` is roughly 200 MB.

The `-H` flag suppresses column headings from the output.

Example

The following example shows disk space used in the `sample` directory in `dgroup1`, including all of the directories under the `sample` directory.

```
ASMCMD [+dgroup1/sample] > du
Used_MB      Mirror_used_MB
   1251                2507
```

exit Command

Purpose

Exits ASMCMD and returns control to the operating system command-line prompt.

Syntax

```
exit
```

find Command

Purpose

Displays the absolute paths of all occurrences of the specified name pattern (with wildcards) in a specified directory and its subdirectories.

Syntax and Description

```
find [-t type] dir_name name_pattern
```

This command searches the specified directory and all subdirectories under it in the directory tree for the supplied *`name_pattern`*. The value that you use for *`name_pattern`* can be a directory name or a filename, and can include wildcard characters. See "Wildcard Characters" on page 7-4.

In the output of the command, directory names are suffixed with the slash character (/) to distinguish them from filenames.

You use the `-t` flag to find all the files of a particular type (specified as *`type`*). For example, you can search for control files by specifying *`type`* as `CONTROLFILE`. Valid values for *`type`* are the following:

```
CONTROLFILE
DATAFILE
```

```
ONLINELOG
ARCHIVELOG
TEMPFILE
BACKUPSET
DATAFILE
PARAMETERFILE
DATAGUARDCONFIG
FLASHBACK
CHANGETRACKING
DUMPSET
AUTOBACKUP
XTRANSPORT
```

These are the type values from the `type` column of the `V$ASM_FILE` view.

Examples

The following example searches the `dgroup1` disk group for files that begin with '`UNDO`'.

```
ASMCMD> find +dgroup1 undo*
+dgroup1/sample/DATAFILE/UNDOTBS1.258.555341963
+dgroup1/sample/DATAFILE/UNDOTBS1.272.557429239
```

The following example returns the absolute path of all the control files in the `+dgroup1/sample` directory.

```
ASMCMD> find -t CONTROLFILE +dgroup1/sample *
+dgroup1/sample/CONTROLFILE/Current.260.555342185
+dgroup1/sample/CONTROLFILE/Current.261.555342183
```

help Command

Purpose

Displays the syntax of a command and a description of the command parameters.

Syntax and Description

`help [`*`command`*`]`

If you do not specify a value for *`command`*, then the `help` command lists all of the ASMCMD commands and general information about using the ASMCMD utility.

Example

The following example displays help text for the mkalias command.

```
ASMCMD> help mkalias
```

ls Command

Purpose

Lists the contents of an ASM directory, the attributes of the specified file, or the names and attributes of all disk groups from the `V$ASM_DISKGROUP_STAT` or `V$ASM_DISKGROUP` view. The default view is `V$ASM_DISKGROUP_STAT`.

Syntax and Description

`ls [-lsdrtLacgH] [`*`name`*`]`

name can be a filename or directory name, including wildcard characters. See "Wildcard Characters" on page 7-4.

If *name* is a directory name, then ASMCMD lists the contents of the directory and depending on flag settings, ASMCMD also lists information about each directory member. Directories are listed with a trailing forward slash (/) to distinguish them from files.

If the value that you enter for *name* is a filename, then ASMCMD lists the file and depending on the flag settings, ASMCMD also lists information about the file. The file must be located in the current directory if the filename is specified with a relative path.

Command flags enable you to modify and customize the output of the command. Table 7–3 lists the flags and their descriptions. For disk group information, this command queries the V$ASM_DISKGROUP_STAT view by default, which can be modified by using the -c and -g flags.

Table 7–3 Flags for the ls command

Flag	Description
(none)	Displays only filenames and directory names.
-l	Displays extended file information, including striping and redundancy information and whether the file was system-generated (indicated by Y under the SYS column) or user-created (as in the case of an alias, indicated by N under the SYS column). When used in the "ls -l +" command, displays directory information. Note that not all possible file attributes or disk group attributes are included. To view the complete set of column values for a file or a disk group, query the V$ASM_FILE and V$ASM_DISKGROUP views.
-s	Displays file space information.
-d	If the value for the *name* argument is a directory, then ASMCMD displays information about that directory, rather than the directory contents. Typically used with another flag, such as the -l flag.
-r	Reverses the sort order of the listing.
-t	Sorts the listing by timestamp (latest first) instead of by name.
-L	If the value for the *name* argument is an alias, then ASMCMD displays information about the file that it references. Typically used with another flag, such as the -l flag.
-a	For each listed file, displays the absolute path of the alias that references it, if any.
-c	Selects from the V$ASM_DISKGROUP view or from the GV$ASM_DISKGROUP view if the -g flag is also specified.
-g	Selects from GV$ASM_DISKGROUP_STAT, or from GV$ASM_DISKGROUP if the -c flag is also specified. GV$ASM_DISKGOUP.INST_ID is included in the output.
-H	Suppresses column headings.

If you specify all of the flags, then the command shows a union of their attributes, with duplicates removed. If you enter ls +, then the command returns information about all disk groups, including information about whether the disk groups are mounted.

Examples

```
ASMCMD [+dgroup1/sample/DATAFILE] > ls
EXAMPLE.269.555342243
SYSAUX.257.555341961
SYSTEM.256.555341961
UNDOTBS1.258.555341963
UNDOTBS1.272.557429239
```

```
USERS.259.555341963


ASMCMD [+dgroup1/sample/DATAFILE] > ls -l
Type      Redund  Striped  Time             Sys  Name
DATAFILE  MIRROR  COARSE   APR 18 19:16:07  Y    EXAMPLE.269.555342243
DATAFILE  MIRROR  COARSE   MAY 09 22:01:28  Y    SYSAUX.257.555341961
DATAFILE  MIRROR  COARSE   APR 19 19:16:24  Y    SYSTEM.256.555341961
DATAFILE  MIRROR  COARSE   MAY 05 12:28:42  Y    UNDOTBS1.258.555341963
DATAFILE  MIRROR  COARSE   MAY 04 17:27:34  Y    UNDOTBS1.272.557429239
DATAFILE  MIRROR  COARSE   APR 18 19:16:07  Y    USERS.259.555341963


ASMCMD [+dgroup1/sample/DATAFILE] > ls -lt
Type      Redund  Striped  Time             Sys  Name
DATAFILE  MIRROR  COARSE   MAY 09 22:01:28  Y    SYSAUX.257.555341961
DATAFILE  MIRROR  COARSE   MAY 05 12:28:42  Y    UNDOTBS1.258.555341963
DATAFILE  MIRROR  COARSE   MAY 04 17:27:34  Y    UNDOTBS1.272.557429239
DATAFILE  MIRROR  COARSE   APR 19 19:16:24  Y    SYSTEM.256.555341961
DATAFILE  MIRROR  COARSE   APR 18 19:16:07  Y    USERS.259.555341963
DATAFILE  MIRROR  COARSE   APR 18 19:16:07  Y    EXAMPLE.269.555342243


ASMCMD [+dgroup1/sample/DATAFILE] > ls -l undo*
Type      Redund  Striped  Time             Sys  Name
DATAFILE  MIRROR  COARSE   MAY 05 12:28:42  Y    UNDOTBS1.258.555341963
DATAFILE  MIRROR  COARSE   MAY 04 17:27:34  Y    UNDOTBS1.272.557429239


ASMCMD [+dgroup1/sample/DATAFILE] > ls -s
Block_Size  Blocks      Bytes       Space  Name
      8192   12801  104865792   214958080  EXAMPLE.269.555342243
      8192   48641  398467072   802160640  SYSAUX.257.555341961
      8192   61441  503324672  1011875840  SYSTEM.256.555341961
      8192    6401   52436992   110100480  UNDOTBS1.258.555341963
      8192   12801  104865792   214958080  UNDOTBS1.272.557429239
      8192     641    5251072    12582912  USERS.259.555341963

ASMCMD [+dgroup1] > ls +dgroup1/sample
CONTROLFILE/
DATAFILE/
ONLINELOG/
PARAMETERFILE/
TEMPFILE/
spfilesample.ora

ASMCMD [+dgroup1] > ls -l +dgroup1/sample
Type  Redund  Striped  Time  Sys Name
                             Y   CONTROLFILE/
                             Y   DATAFILE/
                             Y   ONLINELOG/
                             Y   PARAMETERFILE/
                             Y   TEMPFILE/
                             N   spfilesample.ora=>
                              +dgroup1/sample/PARAMETERFILE/spfile.270.555342443

ASMCMD [+dgroup1] > ls -r +dgroup1/sample
spfilesample.ora
TEMPFILE/
PARAMETERFILE/
```

```
ONLINELOG/
DATAFILE/
CONTROLFILE/

ASMCMD [+dgroup1] > ls -lL example_df2.f
Type      Redund  Striped  Time          Sys  Name
DATAFILE  MIRROR  COARSE   APR 27 11:04  N    example_df2.f =>
+dgroup1/sample/DATAFILE/EXAMPLE.271.556715087

ASMCMD [+dgroup1] > ls -a +dgroup1/sample/DATAFILE/EXAMPLE.271.556715087
+dgroup1/example_df2.f => EXAMPLE.271.556715087

ASMCMD [+dgroup1] > ls -lH +dgroup1/sample/PARAMETERFILE
PARAMETERFILE  MIRROR  COARSE   MAY 04 21:48  Y    spfile.270.555342443

ASMCMD [+dgroup1] > ls -l +
State    Type    Rebal  Unbal  Name
MOUNTED  NORMAL  N      N      DGROUP1/
MOUNTED  NORMAL  N      N      DGROUP2/
MOUNTED  EXTERN  N      N      DGROUP3/
```

lsct Command

Purpose

Lists information about current ASM clients from the `V$ASM_CLIENT` view. A client is a database that uses disk groups that are managed by the ASM instance to which ASMCMD is currently connected.

Syntax and Description

`lsct [-gH] [`*`group`*`]`

If *`group`* is specified, then information about only that disk group is listed. Table 7–4 lists the flags for the `lsct` command.

Table 7–4 Flags for the lsct command

Flag	Description
(none)	Displays information about current ASM clients from the `V$ASM_CLIENT` view.
`-g`	Selects from the `GV$ASM_CLIENT` view. `GV$ASM_CLIENT.INST_ID` is included in the output.
`-H`	Suppresses column headings.

Example

The following example displays information about the client that is accessing the `dgroup1` disk group.

```
ASMCMD [+] > lsct dgroup1
```

lsdg Command

Purpose

Lists all disk groups and their attributes from the `V$ASM_DISKGROUP_STAT` or `V$ASM_DISKGROUP` view. The default view is `V$ASM_DISKGROUP_STAT`. The output also includes notification of any current rebalance operation.

Syntax and Description

```
lsdg [-gcH] [group]
```

If *group* is specified, then information about only that disk group is listed. Table 7–5 lists the flags for the `lsdg` command.

Table 7–5 Flags for the lsdg command

Flag	Description
(none)	Displays the disk group attributes listed in Table 7–6.
`-c`	Selects from `V$ASM_DISKGROUP`, or from `GV$ASM_DISKGROUP` if the -g flag is also specified. This option is ignored if the ASM instance is version 10.1 or earlier.
`-g`	Selects from `GV$ASM_DISKGROUP_STAT`, or from `GV$ASM_DISKGROUP` if the -c flag is also specified. `GV$ASM_DISKGOUP.INST_ID` is included in the output. The `REBAL` column of the `GV$ASM_OPERATION` view is also included in the output.
`-H`	Suppresses column headings.

Table 7–6 shows the attributes for each disk group. To view the complete set of attributes for a disk group, use the `V$ASM_DISKGROUP_STAT` view.

Table 7–6 Attribute descriptions for lsdg command output

Attribute Name	Description
State	Mounted/connected state of the disk group.
Type	Disk group redundancy (`NORMAL`, `HIGH`, `EXTERNAL`).
Rebal	`Y` if a rebalance operation is in progress.
Sector	Sector size in bytes.
Block	Block size in bytes.
AU	Allocation unit size in bytes.
Total_MB	Size of the disk group in MB.
Free_MB	Free space in the disk group in MB, without regard to redundancy. From the `V$ASM_DISKGROUP` view.
Req_mir_free_MB	Amount of space that must be available in the disk group to restore full redundancy after the most severe failure that can be tolerated by the disk group. This is the `REQUIRED_MIRROR_FREE_MB` column from the `V$ASM_DISKGROUP` view.
Usable_file_MB	Amount of free space, adjusted for mirroring, that is available for new files. From the `V$ASM_DISKGROUP` view.
Offline_disks	Number of offline disks in the disk group. Offline disks are eventually dropped.
Name	Disk group name.

Example

The following example lists the attributes of the `dgroup2` disk group.

```
ASMCMD [+] > lsdg dgroup2
```

lsdsk Command

Purpose

List the disks that are visible to ASM, using the V$ASM_DISK_STAT and V$ASM_DISK views. The V$ASM_DISK_STAT view is used by default.

Syntax and Description

```
lsdsk [-ksptagcHI] [-d diskg_roup_name] [pattern]
```

pattern restricts the output to only disks that matches the pattern specified. Wild-card characters and slashes (/ or \) can be part of the pattern. See "Wildcard Characters" on page 7-4.

The k, s, p, and t flags modify how much information is displayed for each disk. If any combination of the flags are specified, then the output shows the union of the attributes associated with each flag. The flags are described in Table 7–7.

This command can run in connected or non-connected mode. The connected mode is always attempted first. The -I option forces the non-connected mode.

- In connected mode, ASMCMD uses dynamic views to retrieve disk information.
- In non-connected mode, ASMCMD scans disk headers to retrieve disk information, using an ASM disk string to restrict the discovery set.

Note: The non-connected mode is not supported on Windows.

Table 7–7 Flags for the lsdsk command

Flag	Description
(none)	Displays the PATH column of the V$ASM_DISK view.
-k	Displays the TOTAL_MB, FREE_MB, OS_MB,NAME, FAILGROUP, LIBRARY, LABEL, UDID, PRODUCT, REDUNDANCY, and PATH columns of the V$ASM_DISK view.
-s	Displays the READS, WRITES, READ_ERRS, WRITE_ERRS, READ_TIME, WRITE_TIME, BYTES_READ, BYTES_WRITTEN, and the PATH columns of the V$ASM_DISK view.
-p	Displays the GROUP_NUMBER, DISK_NUMBER, INCARNATION, MOUNT_STATUS, HEADER_STATUS, MODE_STATUS, STATE, and the PATH columns of the V$ASM_DISK view.
-t	Displays the CREATE_DATE, MOUNT_DATE, REPAIR_TIMER, and the PATH columns of the V$ASM_DISK view.
-g	Selects from GV$ASM_DISK_STAT, or from GV$ASM_DISK if the -c flag is also specified. GV$ASM_DISK.INST_ID is included in the output.
-c	Selects from V$ASM_DISK, or from GV$ASM_DISK if the -g flag is also specified. This option is ignored if the ASM instance is version 10.1 or earlier.
-H	Suppresses column headings.
-I	Scans disk headers for information rather than extracting the information from an ASM instance. This option forces the non-connected mode.
-d	Restricts results to only those disks that belong to the group specified by *disk_group_name*.

Example

```
ASMCMD> lsdsk -k -d DATA *_0001
ASMCMD> lsdsk -s -d DATA *_0001
ASMCMD> lsdsk -t -d DATA *_0001
ASMCMD> lsdsk -C -t -d DATA *_0001
ASMCMD> lsdsk -g -t -d DATA *_0001
```

md_backup Command

Purpose

The MD_BACKUP command creates a backup file containing metadata for one or more disk groups. By default all the mounted disk groups are included in the backup file which is saved in the current working directory. If the name of the backup file is not specified, ASM names the file AMBR_BACKUP_INTERMEDIATE_FILE.

Syntax and Description

md_backup [-b *location_of_backup*] [-g *dgname* [-g *dgname* ...]]

location_of_backup is the location in which you want to store the backup file

dgname is the name of the disk group that you want to back up

Table 7–8 describes the options for the md_backup command.

Table 7–8 Descriptions for the md_backup command options

Option	Description
-b	Specifies the location in which you want to store the intermediate backup file
-g	Specifies the disk group name which is the disk group name that needs to be backed up

Example

The following is an example of the backup command when you run it without options. This example backs up all of the mounted disk groups and creates the backup image in the current working directory.

```
ASMCMD> md_backup
```

The following is an example creates a backup of disk group asmdsk1 and asmdsk2. The backup that this example creates is saved in the /tmp/dgbackup070222 file.

```
ASMCMD> md_backup -b /tmp/dgbackup070222 -g admdsk1 -g asmdsk2
```

md_restore Command

Purpose

This command restores a disk group backup using various options that are described in this section.

For information about restoring a backup file after disk group compatibility settings have been changed, refer to "Considerations When Setting Disk Group Compatibility" on page 4-25.

Syntax and Description

md_restore -b *backup_file* [-i]

[-t (full)|nodg|newdg] [-f *sql_script_file*]

[-g '*diskgroup_name,diskgroup_name,...*']

[-o '*old_diskgroup_name:new_diskgroup_name,...*']

backup_file is the name of the backup script file that you want to restore

sql_script_file is the name of the SQL script that you want to use

diskgroup_name is the name of the disk group

old_diskgroup_name is the name of the old disk group

new_diskgroup_name is the name of the new disk group

Table 7–9 describes the options for the `md_restore` command.

Table 7–9 Descriptions for the md_restore command options

Option	Description
-b	Reads the metadata information from *backup_file*.
-i	Ignore errors. Normally, if `md_restore` encounters an error, it will stop. Specifying this flag ignores any errors.
-t	Specifies the type of disk group to be created: `full` - Create disk group and restore metadata `nodg` - Restore metadata only. `newdg` - Create disk group with a different name and restore metadata; `-o` is required.
-f	Write SQL commands to <sql_script_file instead of executing them.
-g	Select the disk groups to be restored. If no disk groups are defined, then all disk groups will be restored.
-o	Rename disk group *old_diskgroup_name* to *new_diskgroup_name*.

Example

The following example restores the disk group `asmdsk1` from the backup script and creates a copy.

```
ASMCMD> md_restore -t full -g asmdsk1 -i backup_file
```

The following example takes an existing disk group asmdsk1 and restores its metadata.

```
ASMCMD> md_restore -t nodg -g asmdsk1 -i backup_file
```

The following example restores disk group `asmdsk1` completely but the new disk group that is created is called `asmdsk2`.

```
ASMCMD> md_restore -t newdg  -o 'asmdsk1:asmdsk2'  -i backup_file
```

The following example restores from the backup file after applying the overrides defined in the file `override.txt`.

```
ASMCMD> md_restore -t newdg -of override.txt -i backup_file
```

mkalias Command

Purpose

Creates an alias for the specified system-generated filename.

Syntax and Description

`mkalias` *`file alias`*

alias must be in the same disk group as the system-generated file. Only one alias is permitted for each ASM file.

The SQL equivalent of the `mkalias` command is:

ALTER DISKGROUP *dg_name* ADD ALIAS *user_alias* FOR *file*

Example

The following example creates the `sysaux.f` alias for the fully qualified filename `+dgroup1/sample/DATAFILE/SYSAUX.257.555341961`.

```
ASMCMD [+dgroup1/sample/DATAFILE] > mkalias SYSAUX.257.555341961 sysaux.f
ASMCMD [+dgroup1/sample/DATAFILE] > ls -a
none => EXAMPLE.269.555342243
+dgroup1/sample/DATAFILE/sysaux.f => SYSAUX.257.555341961
none => SYSTEM.256.555341961
none => UNDOTBS1.258.555341963
none => UNDOTBS1.272.557429239
none => USERS.259.555341963
sysaux.f
```

mkdir Command

Purpose

Creates ASM directories under the current directory.

Syntax and Description

mkdir *dir_name [dir_name . . .]*

The current directory can be created by the system or by the user. You cannot create a directory at the root (+) level.

The SQL equivalent of the `mkdir` command is:

ALTER DISKGROUP *dg_name* ADD DIRECTORY *dir, dir . . .*

Example

The following example creates the directories `subdir1` and `subdir2` at the disk group level in the disk group `dgroup1`.

```
ASMCMD [+dgroup1] > mkdir subdir1 subdir2
ASMCMD [+dgroup1] > ls
sample/
example_df2.f
subdir1/
subdir2/
```

pwd Command

Purpose

Displays the absolute path of the current directory.

Syntax and Description

pwd

Example

The following example the current directory.

```
ASMCMD> pwd

+dgroup1/sample/controlfile
```

remap Command

Purpose

Repairs a range of physical blocks on a disk. The `remap` command only repairs blocks that have read disk I/O errors. It does not repair blocks that contain corrupted contents, whether or not those blocks can be read. The command assumes a physical block size of `512` bytes and supports all allocation unit sizes (1 to 64 MB).

Syntax and Description

`remap` *`disk_group_name disk_name block_range`*

`disk_group_name` is the name of the disk group in which a disk must be repaired.

`disk_name` is the name of the disk that must be repaired. The name is listed in the `NAME` column of the `V$ASM_DISK` view.

`block_range` is a range of physical blocks to repair, in the format:

`start_range_number-end_range_number`

Example

The following example repairs blocks `5000` through `5999` for disk `DATA_0001` in disk group `DISK_GRP1`.

```
ASMCMD> remap DISK_GRP1 DATA_0001 5000-5999
```

The following example repairs blocks `6230` through `6339` for disk `largedisk_2` in disk group `DISK_GRP2`.

```
ASMCMD> remap DISK_GRP2 largedisk_2 6230-6339
```

rm Command

Purpose

Deletes the specified ASM files and directories.

Syntax and Description

`rm [-rf]` *name [name] . . .*

If *`name`* is a file or alias, then the `rm` command can delete the file or alias only if it is not currently in use by a client database. If *`name`* is a directory, then the rm command can delete it only if it is empty (unless the `-r` flag is used) and it is not a system-generated directory. If *`name`* is an alias, then the `rm` command deletes both the alias and the file to which the alias refers. To delete only an alias and retain the file that the alias references, use the `rmalias` command.

The SQL equivalents of the `rm` command are:

```
ALTER DISKGROUP ... DROP FILE
ALTER DISKGROUP ... DROP DIRECTORY
```

name can contain wildcard characters. See "Wildcard Characters" on page 7-4.

If you use a wildcard, the `rm` command deletes all of the matches except nonempty directories, unless you use the `-r` flag. To recursively delete, use the `-r` flag. This enables you to delete a nonempty directory, including all files and directories in it and in the entire directory tree underneath it. If you use the `-r` flag or a wildcard character, then the `rm` command prompts you to confirm the deletion before proceeding, unless you specify the `-f` flag.

If a wildcard character matches an alias or a system-generated file that has an alias, then both the alias and the system-generated file that it references are deleted. When using the `-r` flag, either the system-generated file or the alias must be present in the directory in which you run the `rm` command.

For example, if you have a user alias, `+dg1/dir1/file.alias` that points to `+dg/orcl/DATAFILE/System.256.146589651`, then running the `rm -r +dg1/dir1` command removes the `+dg1/dir1/file.alias` as well as `+dg/orcl/DATAFILE/System.256.146589651`.

Example

The following example deletes the alias `alias293.f`.

```
ASMCMD [+dgroup1/sample/DATAFILE] > rm alias293.f
```

rmalias Command

Purpose

Deletes the specified aliases, retaining the files that the aliases reference.

Syntax and Description

`rmalias [-r]` *alias [alias] . . .*

To recursively delete, use the `-r` flag. This enables you to delete all of the aliases in the current directory and in the entire directory tree beneath the current directory. If any user-created directories become empty as a result of deleting aliases, they are also deleted. Files and directories created by the system are not deleted.

The SQL equivalent of the `rmalias` command is:

`ALTER DISKGROUP` *`dg_name`* `DROP ALIAS` *`user_alias`*

Example

The following example deletes the alias `sysaux.f`, retaining the datafile that it references.

```
ASMCMD [+dgroup1/sample/DATAFILE] > rmalias sysaux.f
```

A

Deploying ASM Using SQL*Plus

This appendix describes how to deploy Automatic Storage Management (ASM) using SQL*Plus under the following topics:

- Creating a Database that Uses ASM
- Creating Tablespaces in ASM
- Creating Redo Logs in ASM
- Creating Control Files in ASM
- Creating Archive Log Files in ASM

> **See Also:** *Oracle Database Administrator's Guide* for information about using Oracle Managed Files (OMF)

Creating a Database that Uses ASM

The recommended method of creating your database is to use the Database Configuration Assistant (DBCA). However, if you create your database manually using the CREATE DATABASE statement, then ASM enables you to create a database and all of its underlying files with a minimum of input.

The following section describes an example of using the CREATE DATABASE statement, where database files are created and managed automatically by ASM.

Creating a Database that Uses ASM: Example

This example creates a database with the following ASM files:

- A SYSTEM tablespace datafile in disk group dgroup1.
- A SYSAUX tablespace datafile in disk group dgroup1. The tablespace is locally managed with automatic segment-space management.
- A multiplexed online redo log with two online log groups, one member of each in dgroup1 and dgroup2 (flash recovery area).
- If automatic undo management mode is enabled, then an undo tablespace datafile in directory dgroup1.
- If you do not specify a value for the CONTROL_FILES initialization parameter, then two control files, one in drgoup1 and another in dgroup2 (flash recovery area). The control file in dgroup1 is the primary control file.

The following initialization parameter settings are included in the initialization parameter file:

```
DB_CREATE_FILE_DEST = '+dgroup1'
DB_RECOVERY_FILE_DEST = '+dgroup2'
DB_RECOVERY_FILE_DEST_SIZE = 10G
```

The following statement is issued at the SQL prompt:

```
CREATE DATABASE sample;
```

Creating Tablespaces in ASM

When ASM creates a datafile for a permanent tablespace (or a temporary file for a temporary tablespace), the datafile is set to auto-extensible with an unlimited maximum size and 100 MB default size. You can use the AUTOEXTEND clause to override this default extensibility and the SIZE clause to override the default size.

ASM applies attributes to the datafile, as specified in the system default template for a datafile as shown in the table in "Managing Disk Group Templates" on page 5-15. You can also create and specify your own template.

Files in a tablespace might be in both ASM files and non-ASM files as a result of the tablespace history. RMAN commands enable non-ASM files to be relocated to an ASM disk group and enable ASM files to be relocated as non-ASM files.

For example, if there is a disk group named dgroup3, you can create a tablespace tblspace3 in that disk group with the following SQL statement:

```
CREATE TABLESPACE tblspace3 DATAFILE '+DGROUP3';
```

Creating a Tablespace in ASM: Out-of-the-Box Example

This example illustrates out-of-the-box usage of ASM. This example lets ASM create and manage the tablespace datafile for you, using Oracle supplied defaults that are adequate for most situations.

Assume the following initialization parameter setting:

```
DB_CREATE_FILE_DEST = '+dgroup2'
```

The following statement creates the tablespace and its datafile:

```
CREATE TABLESPACE tspace2;
```

Creating Redo Logs in ASM

Online redo logs can be created in multiple disk groups, either implicitly in an initialization parameter file or explicitly in an ALTER DATABASE...ADD LOGFILE statement. Each online log should have one log member in multiple disk groups. The filenames for log file members are automatically generated. All partially created redo log files, created as a result of a system error, are automatically deleted.

For example, if there is are disk groups named dgroup3 and dgroup4, you can create a log file with a member in each disk group with the following SQL statement:

```
ALTER DATABASE ADD LOGFILE ('+DGROUP3','+DGROUP4');
```

Adding New Redo Log Files: Example

The following example creates a log file with a member in each of the disk groups dgroup1 and dgroup2. The following parameter settings are included in the initialization parameter file:

```
DB_CREATE_ONLINE_LOG_DEST_1 = '+dgroup1'
DB_CREATE_ONLINE_LOG_DEST_2 = '+dgroup2'
```

The following statement is issued at the SQL prompt:

```
ALTER DATABASE ADD LOGFILE;
```

Creating Control Files in ASM

Control files can be explicitly created in multiple disk groups. The filenames for control files are automatically generated. If an attempt to create a control file fails, then Oracle deletes partially created control automatically.

If you need to specify a control file by name, then you can use alias filenames to reference ASM files and use easy to understand names. Using aliases in the specification of a control file during its creation enables you to later refer to the control file with a meaningful name. Furthermore, an alias can be specified as a control file name in the CONTROL_FILES initialization parameter. Control files that you create without aliases can be given alias names at a later time using the ALTER DISKGROUP...CREATE ALIAS statement.

When creating a control file, datafiles and log files that are stored in an ASM disk group should be given to the CREATE CONTROLFILE command using the file reference context form of their ASM filenames. However, the use of the RESETLOGS option requires the use of a file creation context form for the specification of the log files.

You can specify a disk group name (or any ASM file creation context) as part of the CONTROL_FILES parameter prior to database creation to have the control files managed by ASM.

Creating Control Files in ASM: Example 1

The following CREATE CONTROLFILE statement is generated by an ALTER DATABASE BACKUP CONTROLFILE TO TRACE command for a database with datafiles and log files created on disk groups dgroup1 and dgroup2:

```
CREATE CONTROLFILE REUSE DATABASE "SAMPLE" NORESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 2
    MAXDATAFILES 30
    MAXINSTANCES 1
    MAXLOGHISTORY 226
LOGFILE
  GROUP 1 (
    '+DGROUP1/db/onlinelog/group_1.258.541956457',
    '+DGROUP2/db/onlinelog/group_1.256.541956473'
  ) SIZE 100M,
  GROUP 2 (
    '+DGROUP1/db/onlinelog/group_2.257.541956477',
    '+DGROUP2/db/onlinelog/group_2.258.541956487'
  ) SIZE 100M
DATAFILE
  '+DGROUP1/db/datafile/system.260.541956497',
  '+DGROUP1/db/datafile/sysaux.259.541956511'
CHARACTER SET US7ASCII
;
```

Creating Control Files in ASM: Example 2

This example is a CREATE CONTROLFILE statement for a database with datafiles, but uses a RESETLOGS clause, and thus uses the creation context form for log files:

```
CREATE CONTROLFILE REUSE DATABASE "SAMPLE" RESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 2
    MAXDATAFILES 30
    MAXINSTANCES 1
    MAXLOGHISTORY 226
LOGFILE
  GROUP 1 (
    '+DGROUP1',
    '+DGROUP2'
  ) SIZE 100M,
  GROUP 2 (
    '+DGROUP1',
    '+DGROUP2'
  ) SIZE 100M
DATAFILE
  '+DGROUP1/db/datafile/system.260.541956497',
  '+DGROUP1/db/datafile/sysaux.259.541956511'
CHARACTER SET US7ASCII
;
```

Creating Archive Log Files in ASM

Disk groups can be specified as archive log destinations in the LOG_ARCHIVE_DEST and LOG_ARCHIVE_DEST_n initialization parameters. When destinations are specified in this manner, the archive log filename will be unique, even if archived twice. All partially created archive files, created as a result of a system error, are automatically deleted.

If LOG_ARCHIVE_DEST is set to a disk group name, LOG_ARCHIVE_FORMAT is ignored. Unique filenames for archived logs are automatically created by the Oracle database. If LOG_ARCHIVE_DEST is set to a directory in a disk group, LOG_ARCHIVE_FORMAT has its normal semantics.

The following sample archive log names might be generated with DB_RECOVERY_FILE_DEST set to +dgroup2. SAMPLE is the value of the DB_UNIQUE_NAME parameter:

```
+DGROUP2/SAMPLE/ARCHIVELOG/2003_09_23/thread_1_seq_38.614.541956473
+DGROUP2/SAMPLE/ARCHIVELOG/2003_09_23/thread_4_seq_35.609.541956477
+DGROUP2/SAMPLE/ARCHIVELOG/2003_09_23/thread_2_seq_34.603.541956487
+DGROUP2/SAMPLE/ARCHIVELOG/2003_09_25/thread_3_seq_100.621.541956497
+DGROUP2/SAMPLE/ARCHIVELOG/2003_09_25/thread_1_seq_38.614.541956511
```

Glossary

ASM Cluster

An ASM cluster is a collection of interconnected nodes, each with an ASM instance, operating as a unified cluster with using Oracle Clusterware. An ASM cluster presents a shared pool of storage to one or more Oracle Databases that are also operating on the nodes. The databases can also be clustered using Oracle Real Application Clusters, but that is not a requirement. The disks that provide the shared storage pool must be accessible from all of the nodes in the ASM cluster.

ASMLIB

ASMLIB is an application programming interface (API) developed by Oracle to simplify the operating system–to-database interface and to exploit the capabilities of vendors' storage arrays on Linux-based systems.

Cluster File System (CFS)

A Cluster File System, or CFS, is a file system that is distributed across multiple computer systems. Generally, the computer systems share access to a disk connected through a **Storage Area Network (SAN)**. The CFS component on each individual computer system coordinates access to the disks using a global communication mechanism.

data extent

A data extent is the raw storage used to hold the contents of a file. Each data extent consists of one or more allocation units (AU) on a specific disk.

disk group

An ASM disk group is a collection of disks that ASM manages as a unit. Within a disk group, ASM exposes a file system interface for Oracle Database files. The content of files that are stored in a disk group are evenly distributed, or striped, to eliminate hot spots and to provide uniform performance across the disks. ASM files may also be optionally mirrored within a disk group. The performance of disks in a disk group is comparable to the performance of raw devices.

Direct Attached Storage (DAS)

Direct Attached Storage, or DAS, consists of storage devices that attach directly to host without an intervening network. DAS generally costs less than SAN attached storage, but also offers less flexibility and functionality. Examples of DAS include SATA disks, common to most desktop systems, and SCSI disks that are found on many servers.

extent map

An extent map is a list of extent pointers that point to all the data extents of a file. This is the raw storage for the file. Each extent pointer gives the disk and allocation unit of the data extent. For reliability each extent pointer also includes a check byte to ensure it has not been damaged. This is particularly important when using an in-memory copy of the extent map to direct file I/O to a particular disk location.

fibre channel

Fibre channel is a set of standards that define the interface between computer systems and peripherals. The fibre channel interface is a serial bus interface originally designed to supersede the SCSI standard. However, both the fibre channel and SCSI interfaces have evolved independently of each other and have benefited from the existence of the other. Fibre is spelled with **re** rather than an **er** to indicate a distinction from terms such as *fiber optics*. Both copper and fiber optics are commonly used as a media for fibre channel.

file system

A file system is a software component providing structured access to disks. File systems present objects, such as files, to application programs. Access to files is generally specified with a standard API defining operating system calls such as Open/Close and Read/Write that the application program uses for accessing files. File systems are usually provided as a component of an operating system, but may be provided as an independent software component.

file

A file is an object presented to an application program by a **file system**. A file is subdivided into blocks by the file system. A file system typically places what appears to the application program as consecutive blocks, into nonconsecutive locations on disks. The mapping of the file blocks to the locations on disks is kept in what is known as an **extent map**.

host bus adapter (HBA)

A host bus adapter (HBA), or host adapter, connects a host system to other network and storage devices, such as, eSATA, and SCSI devices.

Logical Unit Number (LUN)

In SCSI terminology, a Logical Unit Number, or LUN, is a number representing a device address presented to a computer system. In storage technology, the term LUN is often used to denote a disk presented to a computer system by a storage array.

mirroring

In storage systems, mirroring is a means for protecting the integrity of data by storing copies of data on multiple disks. If a disk fails, then a secondary copy of the data is available on a second or third disk. Generally, mirroring can be deployed in one of two ways. In the most common case, as with a RAID storage array, a fixed amount of physical storage space on a single disk is uniformly copied on multiple disks. Through software on the storage array, the physically mirrored space on multiple devices is presented to the host as a single disk. Another approach to mirroring is to copy fixed segments of a file onto multiple disks. ASM uses the second approach for maximum flexibility.

Network Attached Storage (NAS)

Network Attached Storage, or NAS, comprises a class of systems providing file services to host computers. A device driver in the operating system accesses data using protocols such as NFS or Common Internet File System (CIFS). From the application program's point of view, Network Attached Storage appears as a file system local to the host computer on which the application program resides.

Network File System (NFS)

A network file system, or NFS, is an industry standard providing Network Attached Storage on many platforms. NFS extends the local file system framework to provide device sharing to users on an IP network. The user view of an NFS is that a remote file system on a host appears as if it were locally mounted.

Oracle Cluster File System (OCFS)

Oracle offers two cluster file systems, OCFS for Windows and OCFS2 for Linux. While OCFS for Windows is a proprietary file system, the source for OCFS2 for Linux is available to all under GNUs' General Public License (GPL). The two file systems are not compatible.

partition

Operating systems typically provide a means for splitting a disk into sections called partitions. To an application, a partition appears as an independent disk, even though that is not the case. The operating system provides a command for managing the size and locations of partitions on a disk. This command writes a partition map onto the disk in an area that is not visible to the application program.

Redundant Array of Inexpensive Disks (RAID)

Redundant Array of Inexpensive Disks, or RAID, is a means for combining multiple disks through software and presenting them to a host as a collection of apparently distinct disks. Mirroring and striping are two means used for combining the disks as a collection. RAID can be implemented as a software component of a computer system or as software operating inside of a storage array. RAID operates on the physical presentation of storage as opposed to ASM mirroring and striping at the logical file level.

Storage Area Network (SAN)

Storage Area Network, or SAN, is a specialized network designed for attaching storage devices, such as disk arrays and tape drives, to computer systems. The most common network infrastructure used in SANs is **fibre channel**. SANs provide great flexibility by allowing a common interconnect infrastructure in which any storage device can be connected to any computer system. Zoning and LUN masking are commonly used for access control for SANs. Zoning is a mechanism where the SAN infrastructure determines the access rights of storage devices to computer systems. LUN masking is a mechanism where the storage device determines the access rights of what computer systems is permitted to connect to the device.

Small Computer System Interface (SCSI)

Small Computer System Interface, or SCSI, is a standard defining the interface between computer systems and peripherals, most commonly storage devices. The SCSI standard defines both the logical and physical aspects of the interface between computer systems and peripherals.

striping

In storage systems, striping is a means for spreading data across multiple disks as opposed to storing the data on a single disk. Striping is usually done to improve performance. Generally, striping can be deployed in one of two ways. In the most common case, as with a RAID storage array, a fixed amount of physical storage space that could have been stored on a single disk is uniformly striped across multiple disks. Through software on the storage array, the physically striped space on multiple devices is presented to the host as a single disk. Another approach to striping is to stripe fixed segments of a file across multiple disks connected to the host. ASM uses the second approach for maximum performance.

volume

In the storage realm, the meaning of volume has many related definitions. Volumes are often thought of as the virtual entity represented as a**Logical Unit Number (LUN)**. Volumes often are presented as an aggregation of pieces from several disks. The mapping of the collection of the pieces of the disks into a volume is managed by a software component called a Volume Manager.

Index