# Oracle 11g DBA Fundamentals Overview

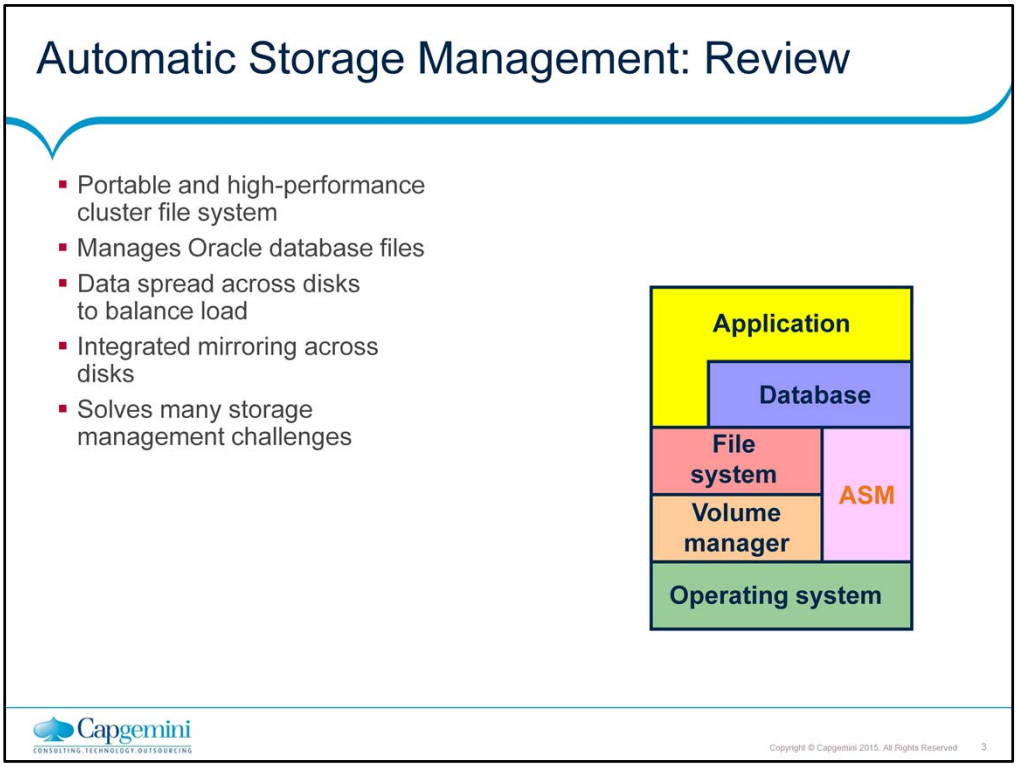
Lesson 08: Using Automatic Storage Management

# Objectives

- After completing this lesson, you should be able to:
  - Identify the features of Automatic Storage Management (ASM)
  - Set up initialization parameter files for ASM and database instances
  - Execute SQL commands with ASM file names
  - Start up and shut down ASM instances
  - Administer ASM disk groups
  - Use RMAN to migrate your database to ASM

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 2



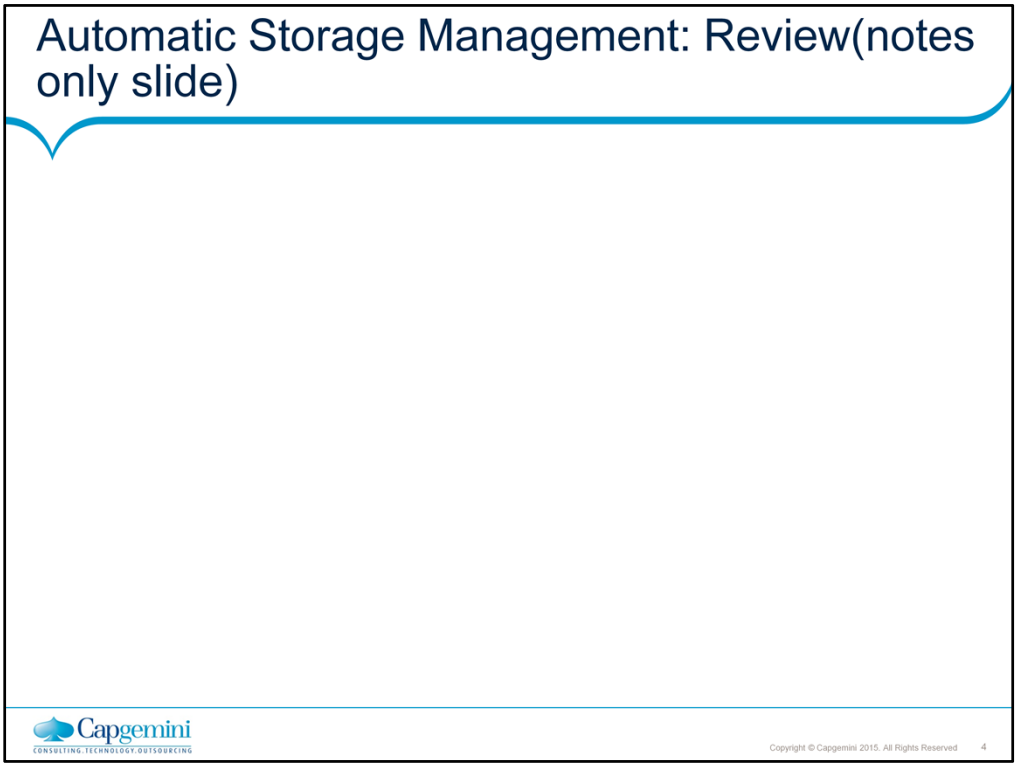

Automatic Storage Management: Review

Automatic Storage Management (ASM) provides a vertical integration of the file system and the volume manager that is specifically built for the Oracle database files. ASM can provide management for single SMP machines, or across multiple nodes of a cluster for Oracle Real Application Clusters (RAC) support.

ASM distributes I/O load across all available resources to optimize performance while removing the need for manual I/O tuning. ASM helps DBAs to manage a dynamic database environment by allowing them to increase the database size without having to shut down the database to adjust the storage allocation.

ASM can maintain redundant copies of data to provide fault tolerance, or it can be built on top of vendor-supplied reliable storage mechanisms. Data management is done by selecting the desired reliability and performance characteristics for classes of data rather than with human interaction on a per file basis.

ASM capabilities save DBAs' time by automating manual storage and thereby increasing their ability to manage larger databases and more of them with increased efficiency.

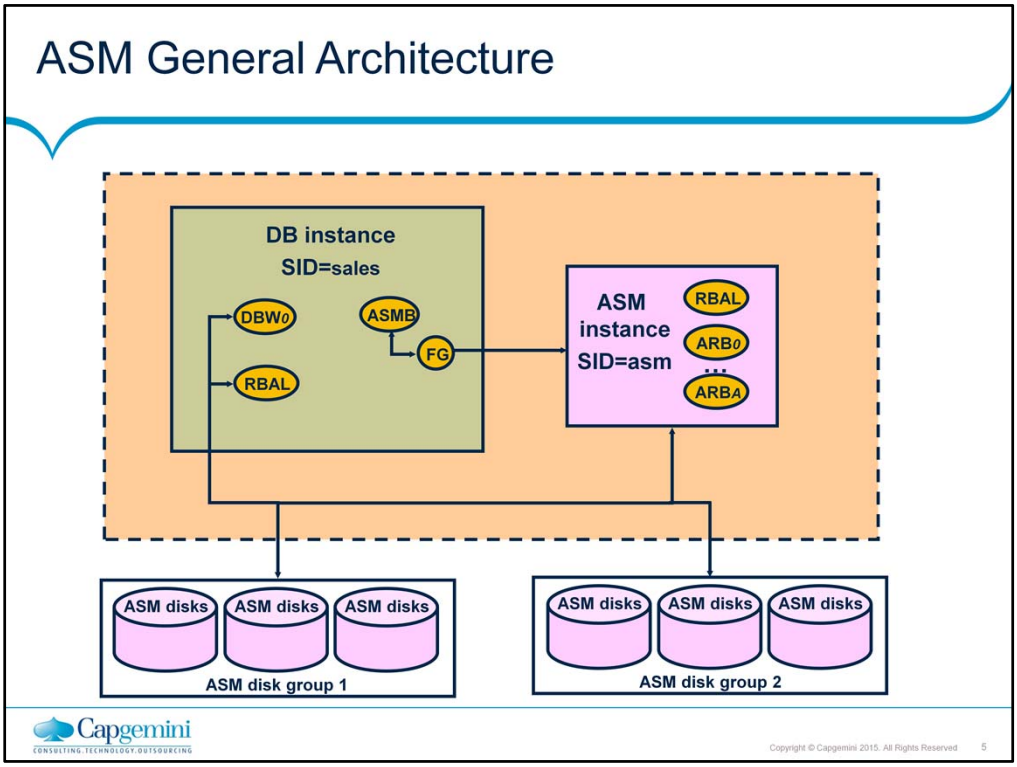



Automatic Storage Management: Review (continued)

ASM divides files into allocation units (AUs) and spreads the AUs for each file evenly across all the disks. ASM uses an index technique to track the placement of each AU. When your storage capacity changes, ASM does not restripe all of the data, but moves an amount of data proportional to the amount of storage added or removed to evenly redistribute the files and maintain a balanced load across the disks. This is done while the database is active.

You can increase the speed of a rebalance operation, or lower it to reduce the impact on the I/O subsystem. ASM provides mirroring protection without the need to purchase a third-party Logical Volume Manager. One unique advantage of ASM is that the mirroring is applied on a file basis, rather than on a volume basis. Therefore, the same disk group can contain a combination of files protected by mirroring, along with those that are not protected at all.

ASM supports data files, log files, control files, archive logs, RMAN backup sets, and other Oracle database file types. ASM supports Real Application Clusters and eliminates the need for a Cluster Logical Volume Manager or a Cluster File System.




ASM General Architecture

To use ASM, you must start a special instance, called an ASM instance, before you start your database instance. ASM instances do not mount databases, instead they manage the metadata needed to make ASM files available to ordinary database instances. Both ASM instances and database instances have access to some common set of disks called disk groups. Database instances access the contents of ASM files directly, communicating with an ASM instance only to get information about the layout of these files.

An ASM instance contains two new background processes. One coordinates rebalance activity for disk groups. It is called RBAL. The second one performs the actual rebalance AU movements. There can be many of these at a time, and they are called ARB0, ARB1, and so forth. An ASM instance also has some of the same background processes as a database instance, including SMON, PMON, LGWR, DBWR, and CKPT.


ASM General Architecture (continued)

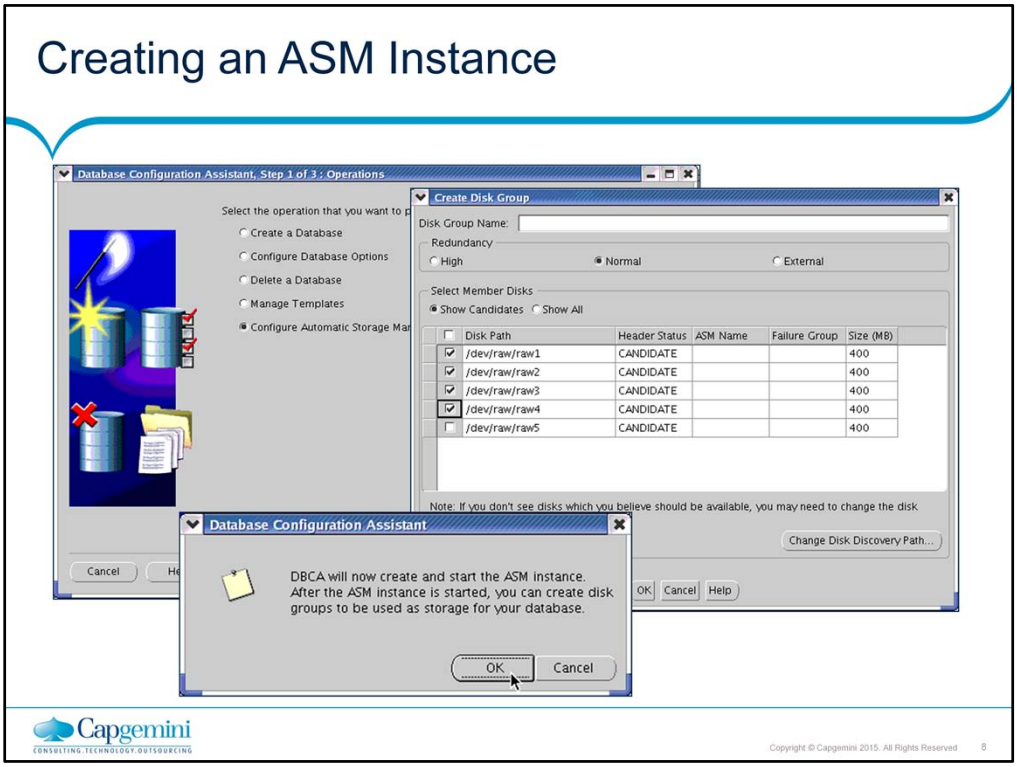
Each database instance using ASM has two new background processes called ASMB and RBAL. RBAL performs global opens of the disks in the disk groups. At database instance startup, ASMB connects as a foreground process into the ASM instance. Communication between the database and the ASM instance is performed via this bridge. This includes physical file changes such as data file creation and deletion. Over this connection, periodic messages are exchanged to update statistics and to verify that both instances are healthy.

## ASM Instance Tasks

- The following are tasks that you need to be able to perform in order to use an ASM instance:
  - Create the ASM instance
  - Set the initialization parameters
  - Start the ASM instance
  - Manage the ASM instance
  - Shut down the ASM instance

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 7



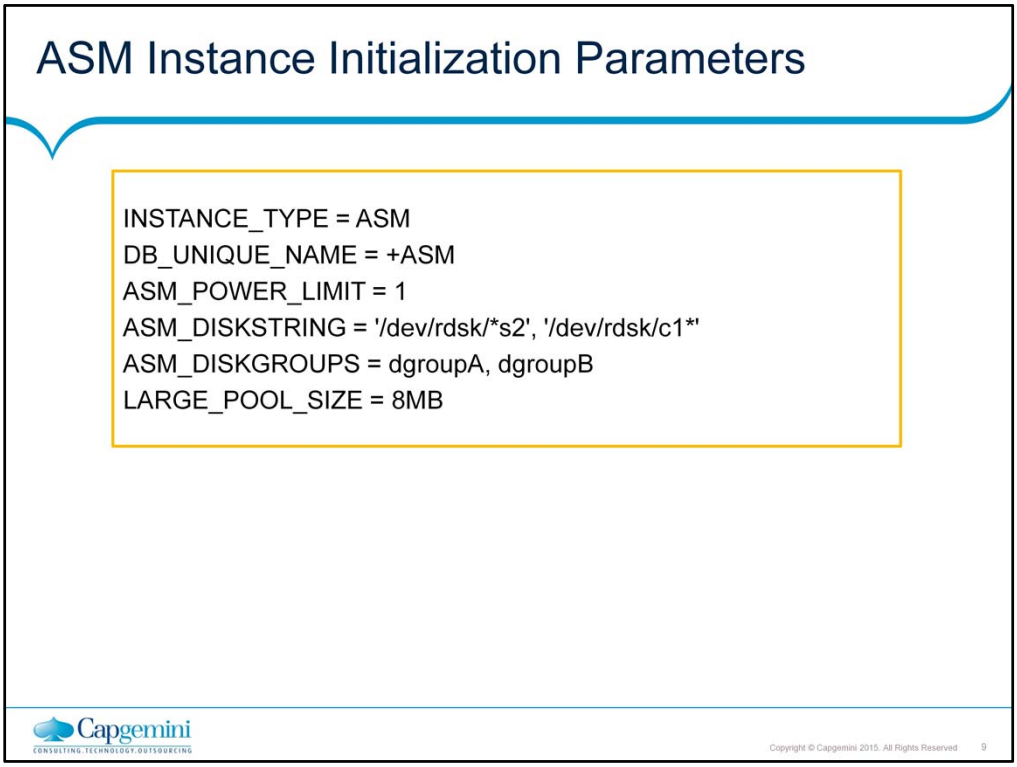

Creating an ASM Instance

You create an ASM instance by running the Database Configuration Assistant (DBCA). On the first screen, choose the option to Configure Automatic Storage Management, and follow the steps. The ASM instance is created and started for you. Then you are guided through the process of defining disk groups for the instance.

As part of the ASM instance creation process, the DBCA automatically creates an entry into the oratab file. This entry is used for discovery purposes. On the Windows platform where a services mechanism is used, the DBCA automatically creates an Oracle Service and the appropriate registry entry to facilitate the discovery of ASM instances.

When an ASM instance is configured, the DBCA creates an ASM instance parameter file and an ASM instance password file.

If you were to first create an ASM-enabled database, the DBCA determines whether an ASM instance already exists on your host. If ASM instance discovery returns an empty list, the DBCA creates a new ASM instance.




ASM Instance Initialization Parameters

INSTANCE_TYPE should be set to ASM for ASM instances.

DB_UNIQUE_NAME specifies the service provider name for which this ASM instance manages disk groups. The default value of +ASM must be modified only if you run multiple ASM instances on the same node.

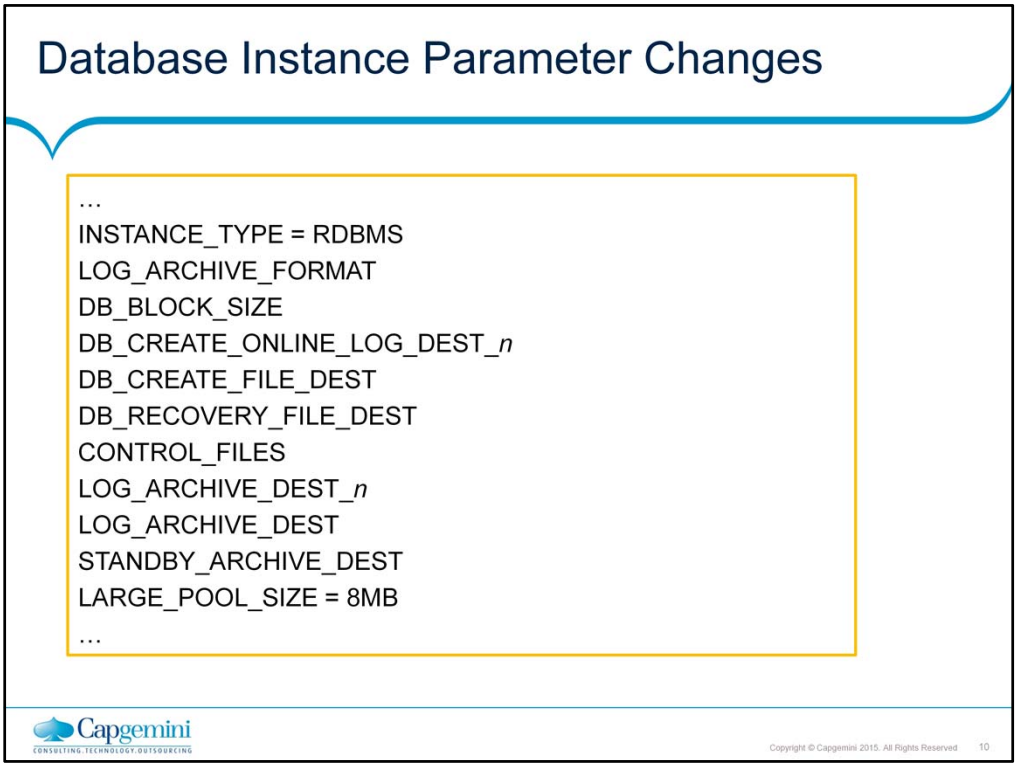
ASM_POWER_LIMIT controls the speed for a rebalance operation. Values range from 1 through 11, with 11 being the fastest. If omitted, this value defaults to 1. The number of slaves is derived from the parallelization level specified in a manual rebalance command (POWER), or by the ASM_POWER_LIMIT parameter.

ASM_DISKSTRING is an operating system–dependent value used by ASM to limit the set of disks considered for discovery.

ASM_DISK_GROUPS is the list of names of disk groups to be mounted by an ASM instance at startup, or when the ALTER DISKGROUP ALL MOUNT command is used.

The INSTANCE_TYPE parameter is the only parameter that you must define. All other ASM parameters have default values that are suitable for most environments.

Note: If the ASM environment has been created using the command line instead of EM, then the disk groups must be created before they can be mounted.




Database Instance Parameter Changes

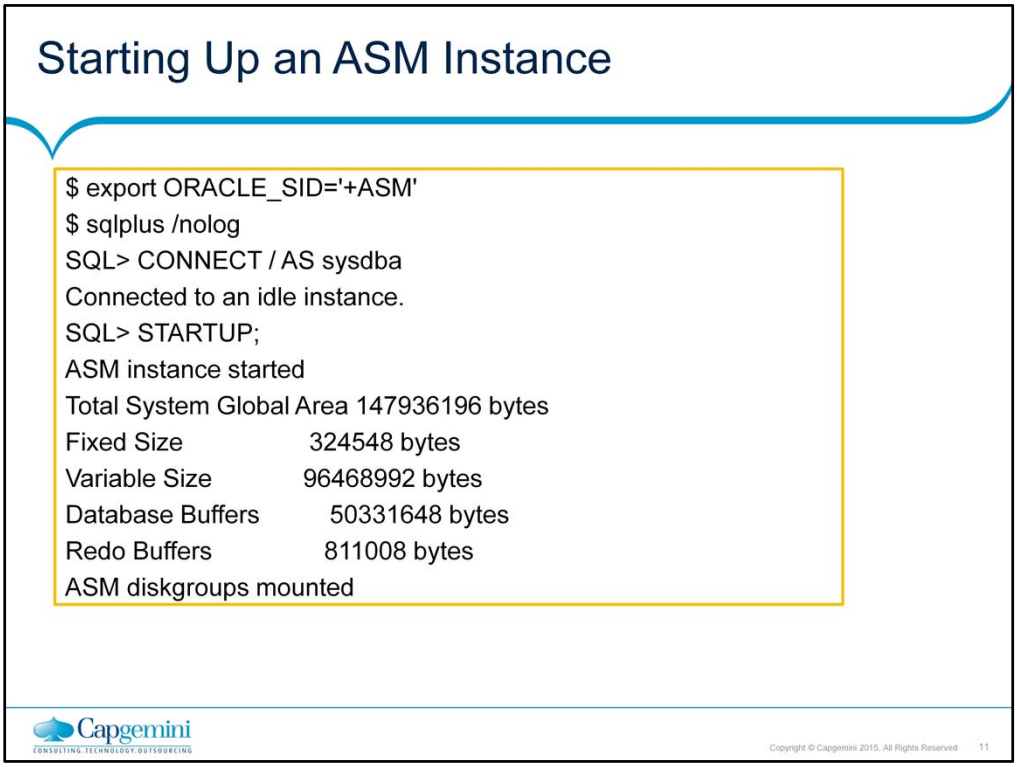
INSTANCE_TYPE defaults to RDBMS and specifies that this instance is an RDBMS instance.

LOG_ARCHIVE_FORMAT is ignored if LOG_ARCHIVE_DEST is set to an incomplete ASM file name, such as +dGroupA. If LOG_ARCHIVE_DEST is set to an ASM directory (for example, +dGroupA/myarchlogdir/), then LOG_ARCHIVE_FORMAT is used and the files are non-OMF. Unique file names for archived logs are automatically created by the Oracle database.

The following parameters accept the multifile creation context form of ASM file names as a destination:

- DB_CREATE_ONLINE_LOG_DEST_n
- DB_CREATE_FILE_DEST
- DB_RECOVERY_FILE_DEST
- CONTROL_FILES
- LOG_ARCHIVE_DEST_n
- LOG_ARCHIVE_DEST
- STANDBY_ARCHIVE_DEST

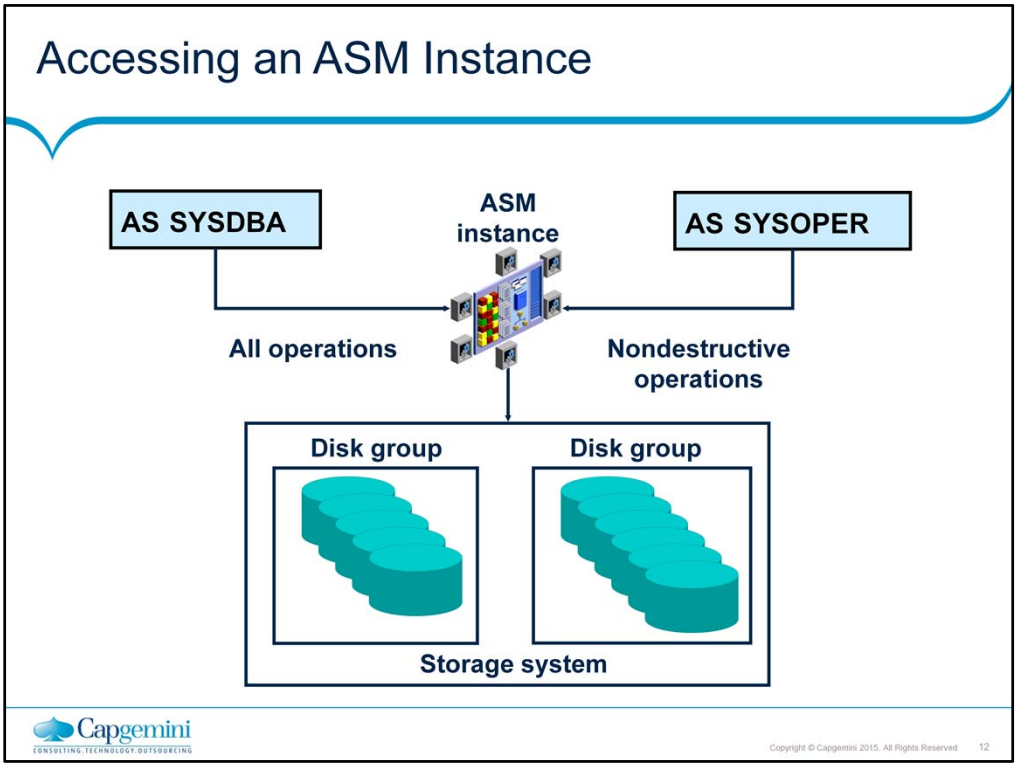
Note: Because allocation unit maps for ASM files are allocated from the LARGE_POOL, you must set the LARGE_POOL_SIZE initialization parameter to at least 8 MB, preferably higher.




Starting Up an ASM Instance

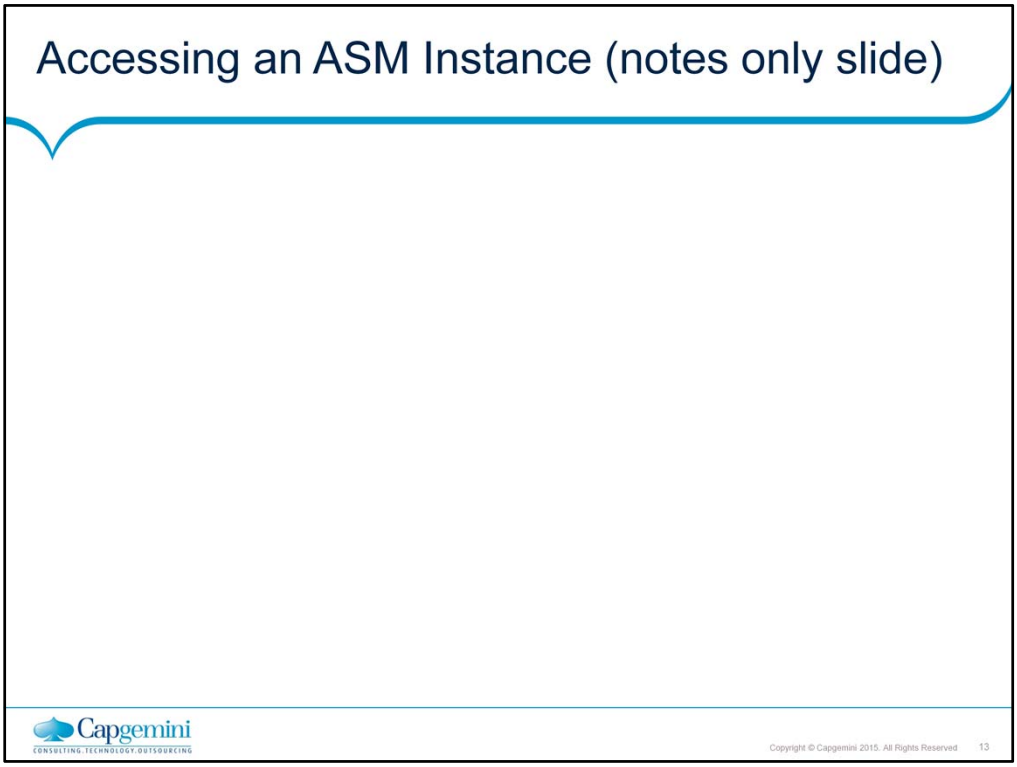
ASM instances are started similarly to database instances except that the initialization parameter file contains an entry like INSTANCE_TYPE=ASM. When this parameter is set to the value ASM, it informs the Oracle executable that an ASM instance is starting, not a database instance. Also, the ORACLE_SID variable must be set to the ASM instance name. When the ASM instance starts up, the mount stage attempts to mount the disk groups specified by the ASM_DISKGROUPS initialization parameter rather than mounting a database, as is done with non-ASM instances.

Other STARTUP clauses have comparable interpretation for ASM instances as they do for database instances. OPEN is invalid for an ASM instance. NOMOUNT starts up the ASM instance without mounting any disk group.




Accessing an ASM Instance

ASM instances do not have a data dictionary, so the only way to connect to one is by using OS authentication, that is, SYSDBA or SYSOPER. To connect remotely, a password file must be used. Normally, the SYSDBA privilege is granted through the use of an operating system group. On UNIX, this is typically the dba group. By default, members of the dba group have SYSDBA privilege on all instances on the node, including the ASM instance. Users who connect to the ASM instance with the SYSDBA privilege have administrative access to all disk groups in the system. The SYSOPER privilege is supported in ASM instances and limits the set of allowable SQL commands to the minimum required for basic operation of an already configured system.

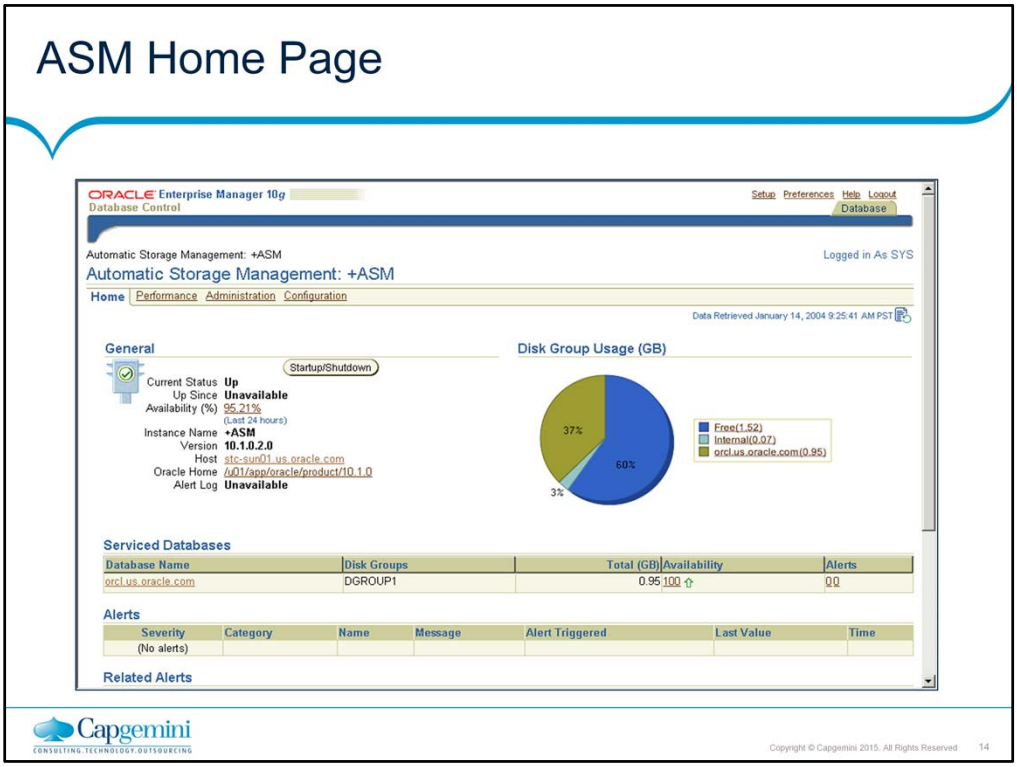



Accessing an ASM Instance (continued)

The following commands are available to SYSOPER users:

- STARTUP/SHUTDOWN
- ALTER DISKGROUP MOUNT/DISMOUNT
- ALTER DISKGROUP ONLINE/OFFLINE DISK
- ALTER DISKGROUP REBALANCE
- ALTER DISKGROUP CHECK
- SELECT all V$ASM_* views

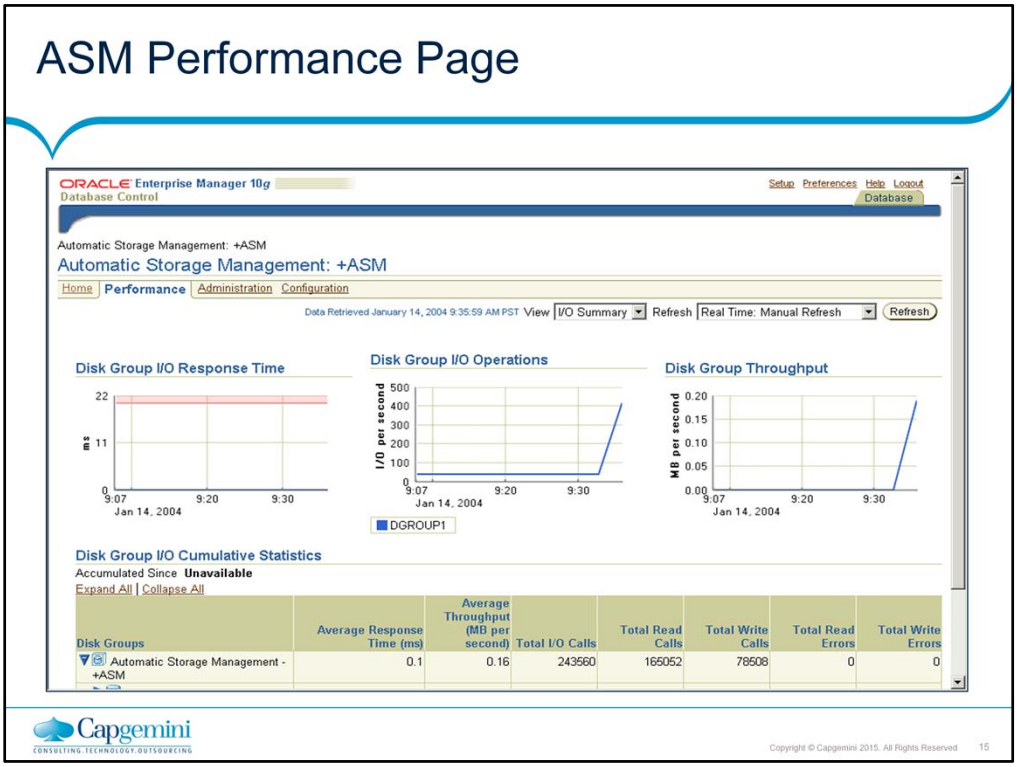
All other commands, such as CREATE DISKGROUP, ADD/DROP/RESIZE DISK, and so on, require the SYSDBA privilege and are not allowed with the SYSOPER privilege.




ASM Home Page

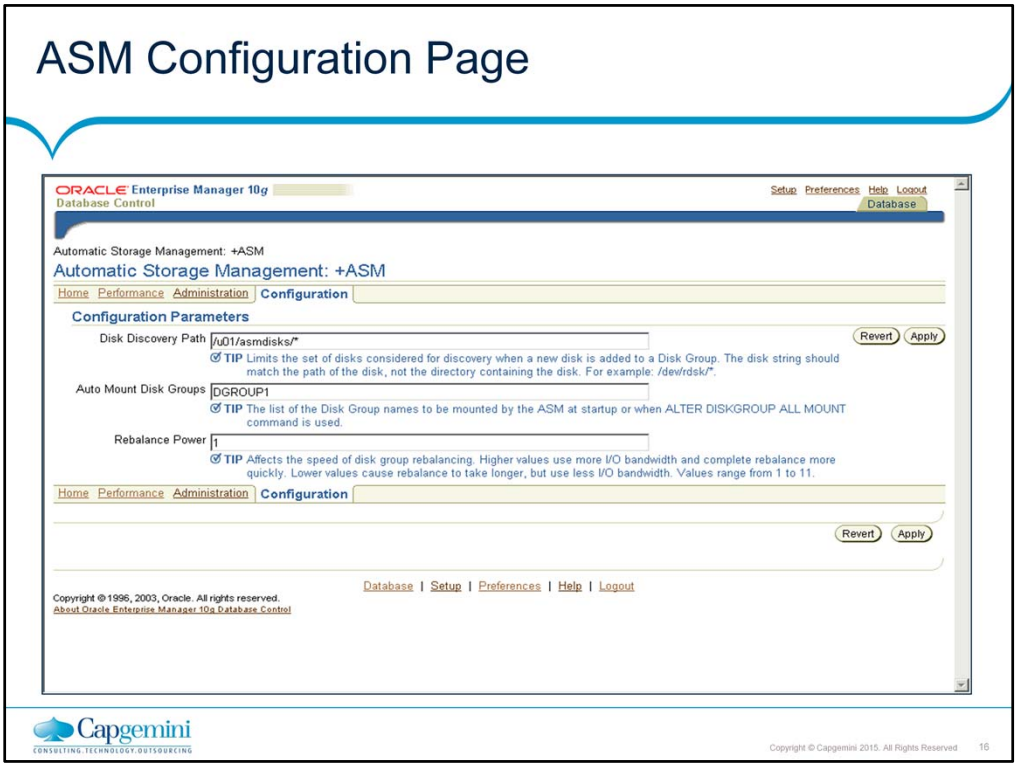
Enterprise Manager provides a user-friendly graphical interface to the Oracle database management, administration, and monitoring tasks. Oracle Database 11g extends the existing functionality to transparently support the management, administration, and monitoring of Oracle databases that use ASM storage. It also adds support for the new management tasks required for administration of ASM instance and ASM disk groups.

The ASM home page shows the status of the ASM instance along with the metrics and alerts generated by the collection mechanisms. It also provides the startup and shutdown functionality. Clicking the Alert link takes the user to an alert details page. The DiskGroup Usage chart shows space used by each client database along with free space.



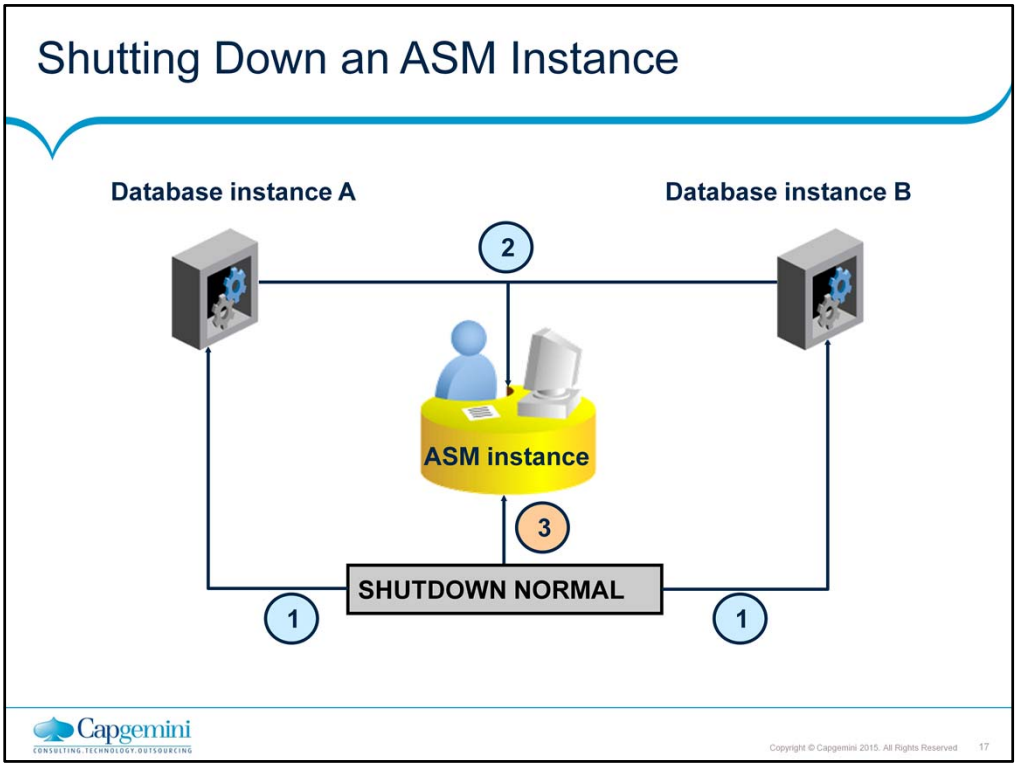

ASM Performance Page

The Performance tab of the Automatic Storage Management page shows the I/O response time and throughput for each disk group. You can further drill down to view disk-level performance metrics.




ASM Configuration Page

The Configuration tab of the Automatic Storage Management page enables you to view or modify the initialization parameters of the ASM instance.




Shutting Down an ASM Instance

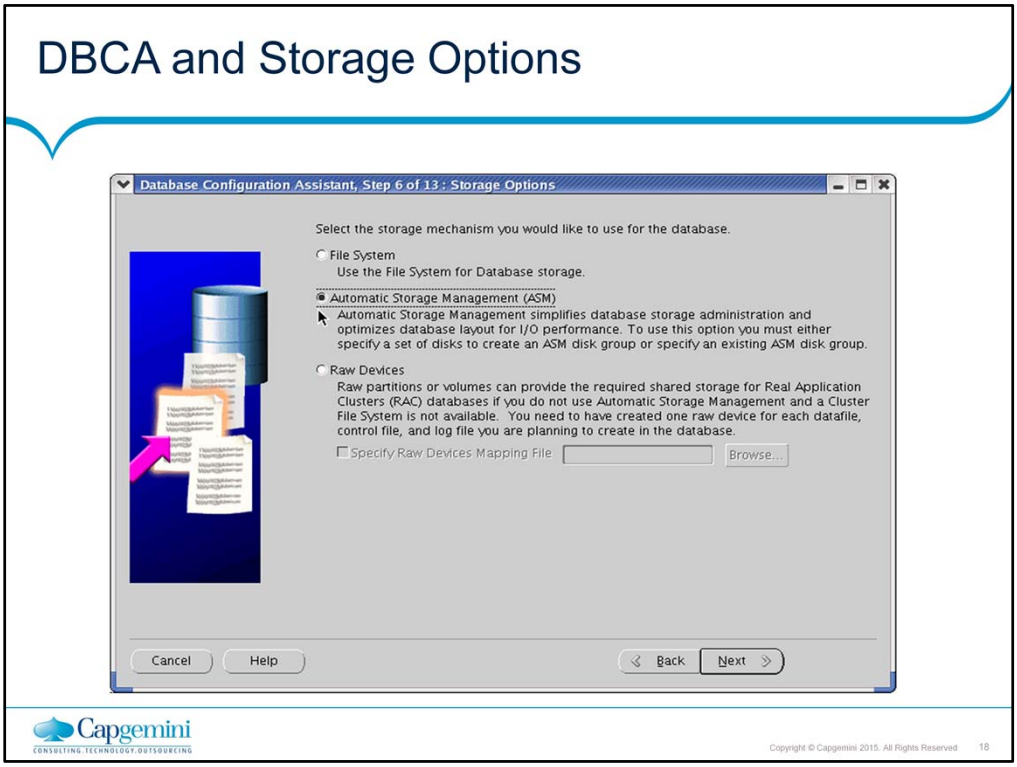
When you attempt to shutdown an ASM instance in the NORMAL, IMMEDIATE, or TRANSACTIONAL modes, it will only succeed if there are no database instances connected to the ASM instance. If there is at least one connected instance, you will receive the following error:

ORA-15097: cannot SHUTDOWN ASM instance with connected RDBMS instance

If you perform a SHUTDOWN ABORT on the ASM instance, it will shutdown, and it will require recovery at the time of the next startup. Any connected database instances will also eventually shutdown, reporting the following error:

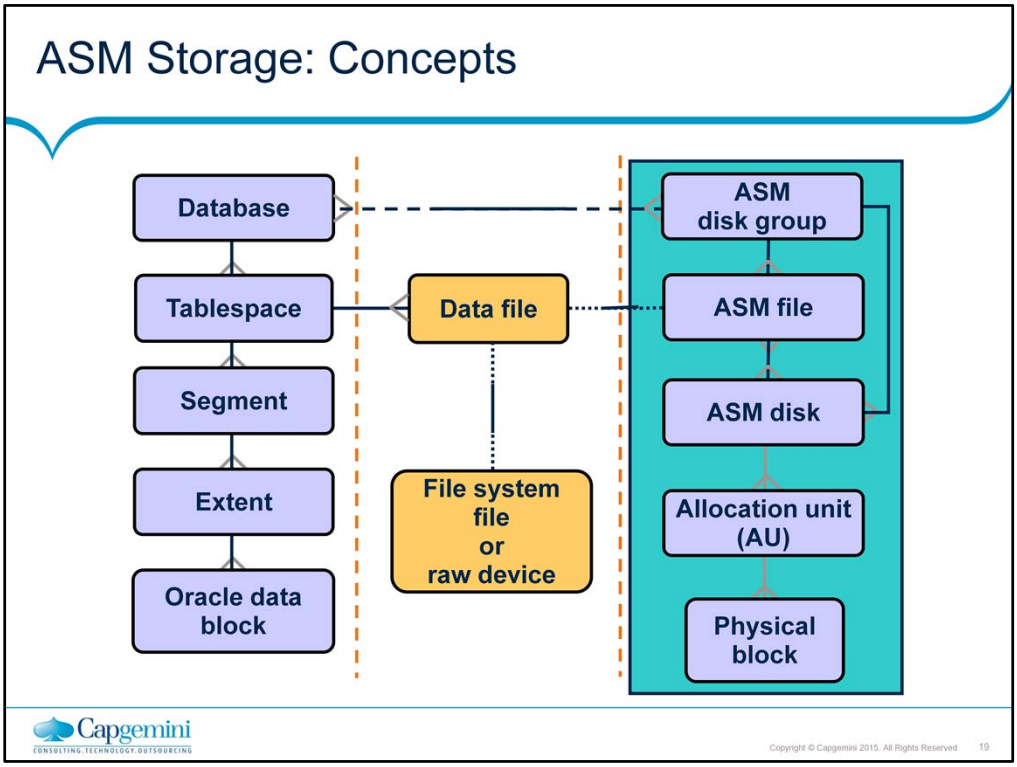
ORA-15064: communication failure with ASM instance

In a single ASM instance configuration, if the ASM instance fails while disk groups are open for update, then after the ASM instance reinitializes, it reads the disk group's log and recovers all transient changes. With multiple ASM instances sharing disk groups, if one ASM instance fails, another ASM instance automatically recovers transient ASM metadata changes caused by the failed instance. The failure of a database instance does not affect ASM instances. The ASM instance should be started automatically whenever the host is rebooted. ASM instance is expected to use the automatic startup mechanism supported by the underlying operating system. Note that file system failure usually crashes a node.




DBCA and Storage Options

In order to support ASM as a storage option, this screen appears in the Database Configuration Assistant (DBCA) when creating a database. This allows you to choose the storage options: file system, ASM, or raw devices.

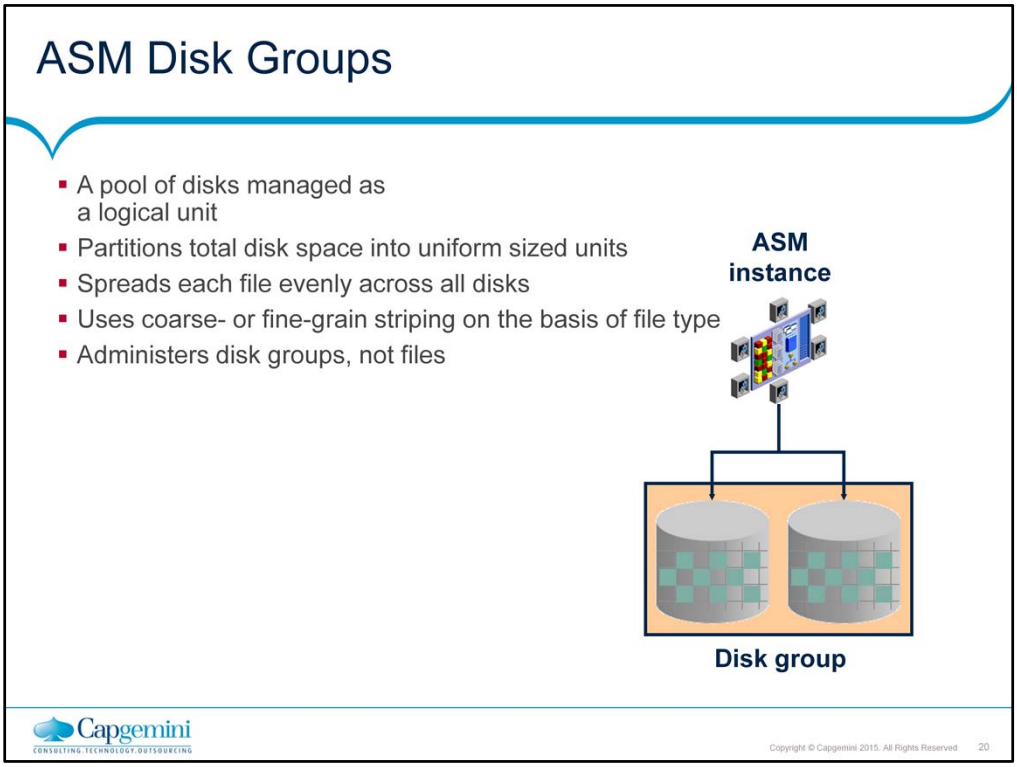



ASM Storage: Concepts

ASM does not eliminate any preexisting database functionality. Existing databases are able to operate as they always have. You can create new files as ASM files and leave existing files to be administered in the old way, or you can eventually migrate them to ASM.

The diagram depicts the relationships that exist between the various storage components inside an Oracle database with ASM available. The left and middle parts of the diagram show the relationships that exist in previous releases. On the right are the new concepts introduced by ASM. Database files can be stored as ASM files. At the top of the new hierarchy are ASM disk groups. Any single ASM file is contained in only one disk group. However, a disk group may contain files belonging to several databases, and a single database may use storage from multiple disk groups. As you can see, one disk group is made up of multiple ASM disks, and each ASM disk belongs to only one disk group. ASM files are always spread across all the ASM disks in the disk group. ASM disks are partitioned in allocation units (AU) of one megabyte each. An allocation unit is the smallest contiguous disk space that ASM allocates. ASM does not allow an Oracle block to be split across allocation units.

Note: This graphic deals with only one type of ASM file: data file. However, ASM can be used to store other database file types.



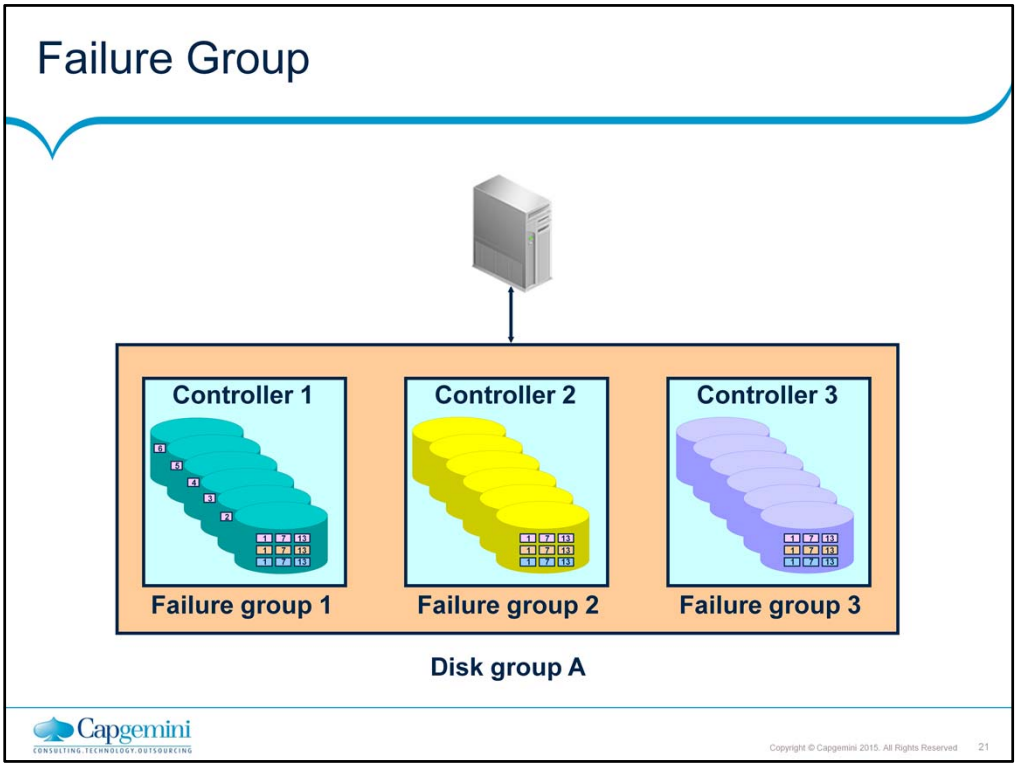

ASM Disk Groups

A disk group is a collection of disks managed as a logical unit. Storage is added and removed from disk groups in units of ASM disks. Every ASM disk has an ASM disk name, which is a name common to all nodes in a cluster. The ASM disk name abstraction is required because different hosts can use different names to refer to the same disk.

ASM always spreads files evenly in 1 MB allocation unit (AU) chunks across all the disks in a disk group. This is called coarse striping. That way, ASM eliminates the need for manual disk tuning. However, disks in a disk group should have similar size and performance characteristics to obtain optimal I/O. For most installations there is only a small number of disk groups. For instance, one disk group for a work area, and one for a recovery area. For files, such as log files, that require low latency, ASM provides fine-grained (128 KB) striping. Fine striping stripes each AU. Fine striping breaks up medium-sized I/O operations into multiple smaller I/O operations that execute in parallel. While the number of files and disks increase, you only have to manage a constant number of disk groups.

From a database perspective, disk groups can be specified as the default location for files created in the database.

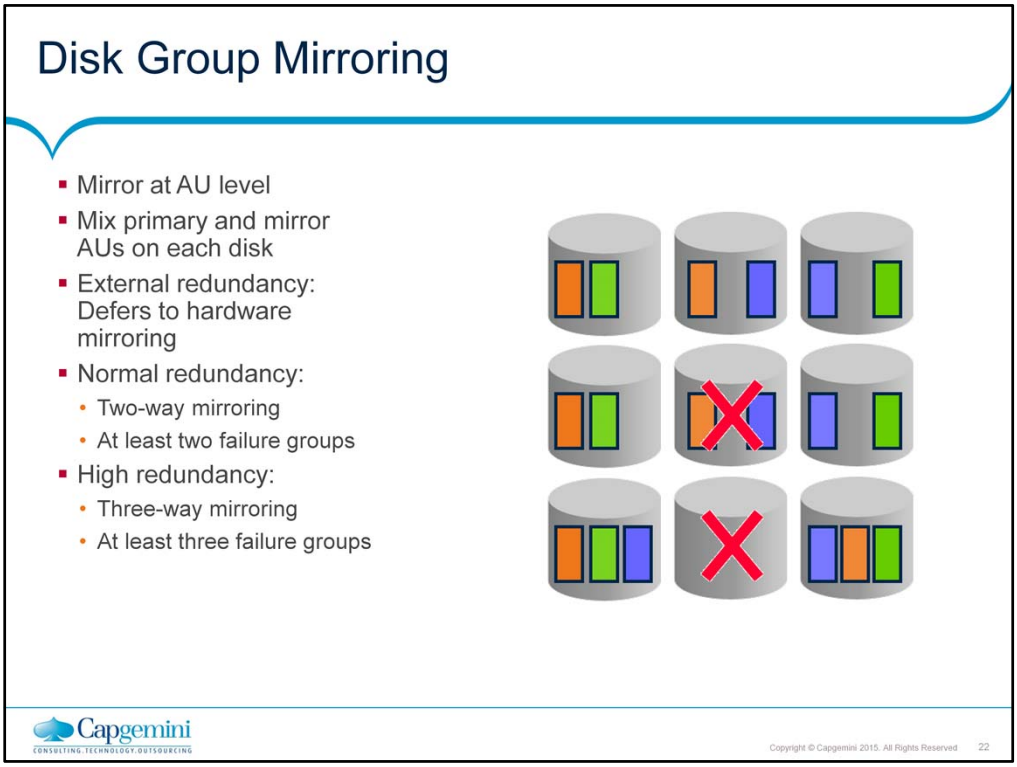
Note: Each disk group is self-describing, containing its own file directory and disk directory.




Failure Group

A failure group is a set of disks, inside one particular disk group, sharing a common resource whose failure needs to be tolerated. An example of a failure group is a string of SCSI disks connected to a common SCSI controller. A failure of the controller leads to all the disks on its SCSI bus becoming unavailable, although each of the individual disks is still functional.

What constitutes a failure group is site-specific. It is largely based upon failure modes that a site is willing to tolerate. By default, ASM assigns each disk to its own failure group. When creating a disk group or adding a disk to a disk group, administrators may specify their own grouping of disks into failure groups. After failure groups are identified, ASM can optimize file layout to reduce the unavailability of data due to the failure of a shared resource.




Disk Group Mirroring

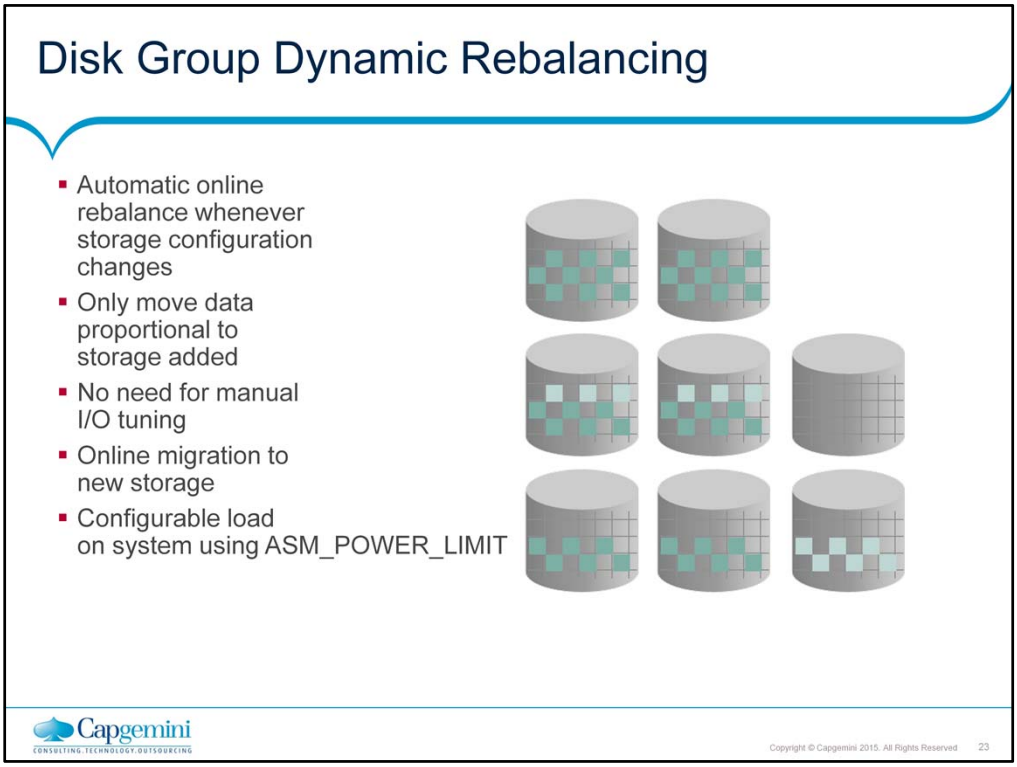
ASM has three disk group types that support different types of mirroring:

External redundancy: Do not provide mirroring. Use an external-redundancy disk group if you use hardware mirroring or if you can tolerate data loss as the result of a disk failure. Failure groups are not used with these types of disk groups.

Normal-redundancy: Support two-way mirroring

High-redundancy: Provide triple mirroring

ASM does not mirror disks; rather, it mirrors allocation units. As a result, you need only spare capacity in your disk group. When a disk fails, ASM automatically reconstructs the contents of the failed disk on the surviving disks in the disk group by reading the mirrored contents from the surviving disks. This spreads the I/O hit from a disk failure across several disks. When ASM allocates a primary AU of a file to one disk in a disk group, it allocates a mirror copy of that AU to another disk in the disk group. Primary AUs on a given disk can have their mirror copies on one of several partner disks in the disk group. ASM ensures that a primary AU and its mirror copy never reside in the same failure group. If you define failure groups for your disk group, ASM can tolerate the simultaneous failure of multiple disks in a single failure group.




Disk Group Dynamic Rebalancing

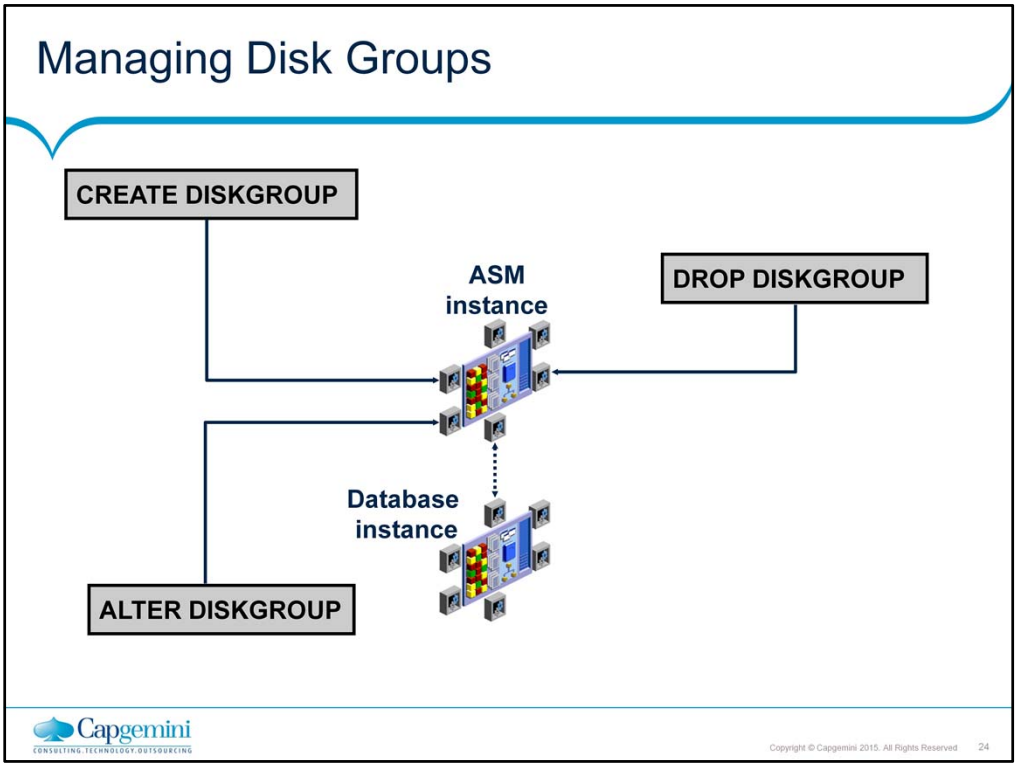
With ASM, the rebalance process is very easy and happens without any intervention from the DBA or system administrator. ASM automatically rebalances a disk group whenever disks are added or dropped.

By using index techniques to spread AUs on the available disks, ASM does not need to restripe all of the data, but instead needs to only move an amount of data proportional to the amount of storage added or removed to evenly redistribute the files and maintain a balanced I/O load across the disks in a disk group.

With the I/O balanced whenever files are allocated and whenever the storage configuration changes, the DBA never needs to search for hot spots in a disk group and manually move data to restore a balanced I/O load.

It is more efficient to add or drop multiple disks at the same time so that they are rebalanced as a single operation. This avoids unnecessary movement of data. With this technique, it is easy to achieve online migration of your data. All you need to do is add the new disks in one operation and drop the old ones in one operation.

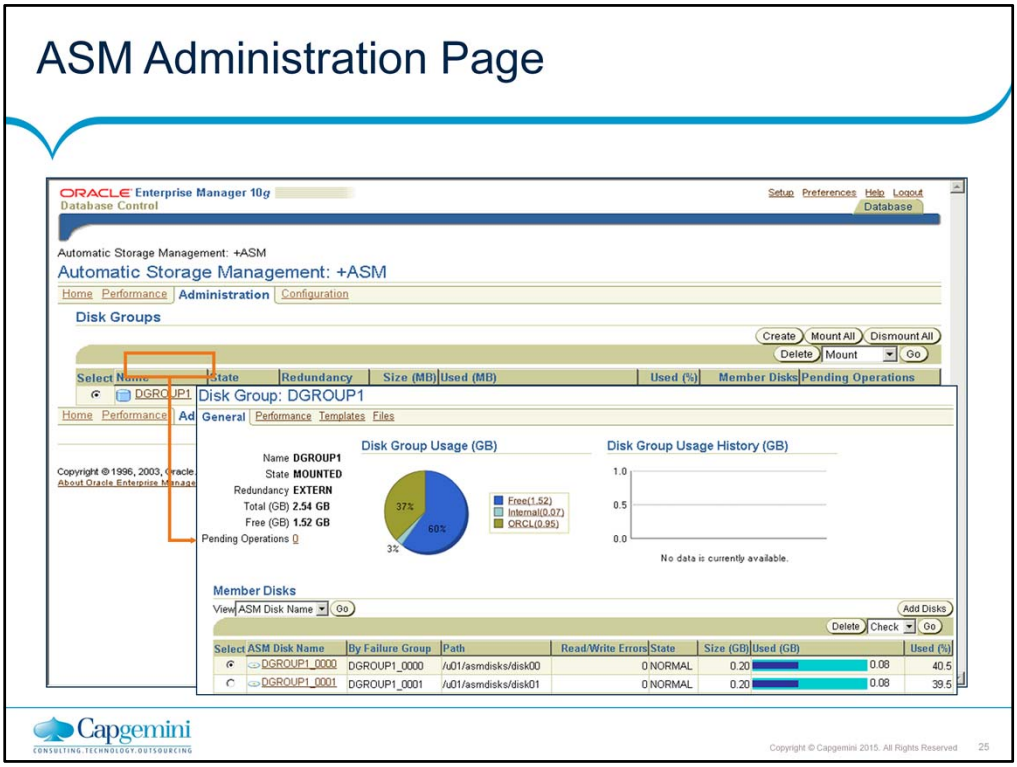
You can control how much of a load the rebalance operation has on the system by setting the ASM_POWER_LIMIT initialization variable. Its range of values is 0 through 11. The lower the number, the lighter the load, whereas a higher setting has more of a load, and finishes sooner. A setting of 0 places rebalance operations on hold. The default value is 1.




Managing Disk Groups

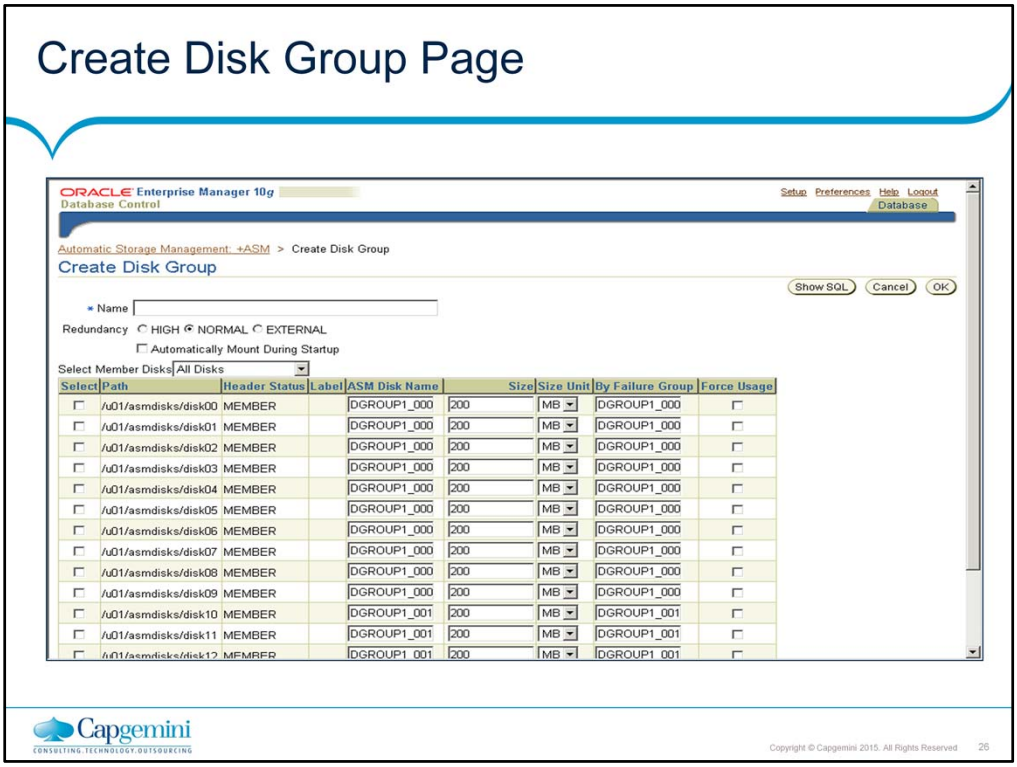
The main goal of an ASM instance is to manage disk groups and protect their data. ASM instances also communicate file layout to database instances. In this way, database instances can directly access files stored in disk groups.

There are several new disk group administrative commands. They all require the SYSDBA privilege and must be issued from an ASM instance. You can add new disk groups. You can also modify existing disk groups to add new disks, remove existing ones, and perform many other operations. You can remove existing disk groups.




ASM Administration Page

When you click the Administration tabbed page of the Automatic Storage Management page, you can see the disk groups listed in the V$ASM_DISKGROUP view. From here, you can create, edit, or drop a disk group. You can also perform disk group operations such as mount, dismount, rebalance, check, and repair on a selected disk group.

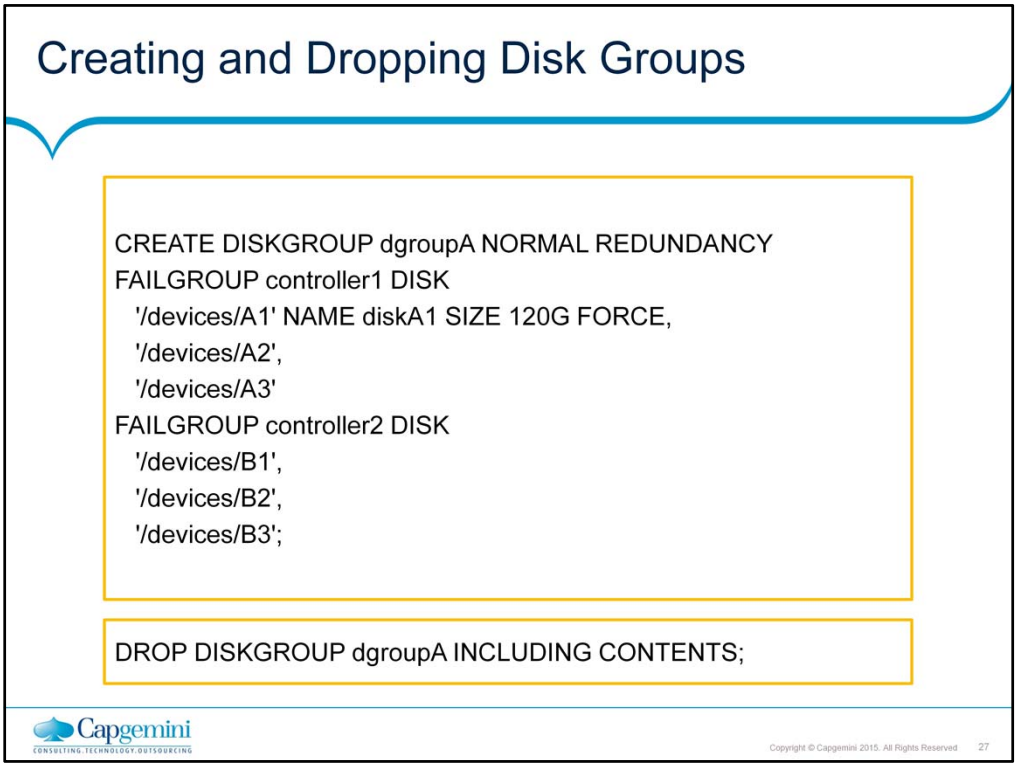



Create Disk Group Page

Clicking Create on the Disk Group Overview page displays the Create Disk Group page. You can enter the disk group name, redundancy mechanism, and the list of disks that you would like to include in the new disk group. The list of disks is obtained from the V$ASM_DISK fixed view. By default, only the disks that can be assigned to a disk group show up. Those are the ones with a status of one of the following:

- CANDIDATE: The disk has never been assigned to an ASM disk group.
- FORMER: The disk was once assigned to an ASM disk group, but is not now.
- PROVISIONED: ASMLib is being used, and this disk is not yet assigned to a disk group.

Note: ASMLib is an API that interfaces with other vendors' storage arrays. See the Database Administrator's Guide documentation for more information about ASMLib.



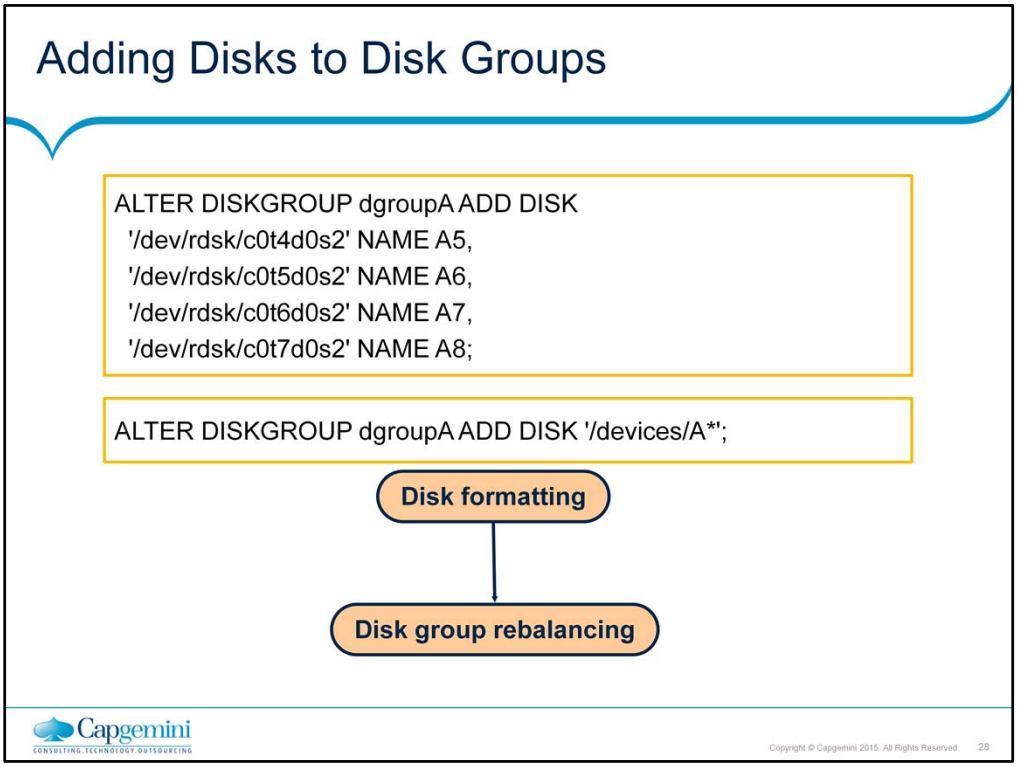

Creating and Dropping Disk Groups

Assume that ASM disk discovery identified the following disks in the /devices directory: A1, A2, A3, B1, B2, and B3. Also, assume that disks A1, A2, and A3 are on a separate SCSI controller from disks B1, B2, and B3.

The first example in the slide illustrates how to configure a disk group called DGROUPA with two failure groups: CONTROLLER1 and CONTROLLER2.

The example also uses the default redundancy characteristic, NORMAL REDUNDANCY, for the disk group. You can optionally provide a disk name and size for the disk. If you do not supply this information, ASM creates a default name and attempts to determine the size of the disk. If the size cannot be determined, an error is returned. FORCE indicates that a specified disk should be added to the specified disk group even though the disk is already formatted as a member of an ASM disk group. Using the FORCE option for a disk that is not formatted as a member of an ASM disk group returns an error.

As shown by the second statement in the slide, you can delete a disk group along with all its files. To avoid accidental deletions, the INCLUDING CONTENTS option must be specified if the disk group still contains any files besides internal ASM metadata. The disk group must be mounted in order for it to be dropped. After ensuring that none of the disk group files are open, the group and all its drives are removed from the disk group. Then the header of each disk is overwritten to eliminate the ASM formatting information.



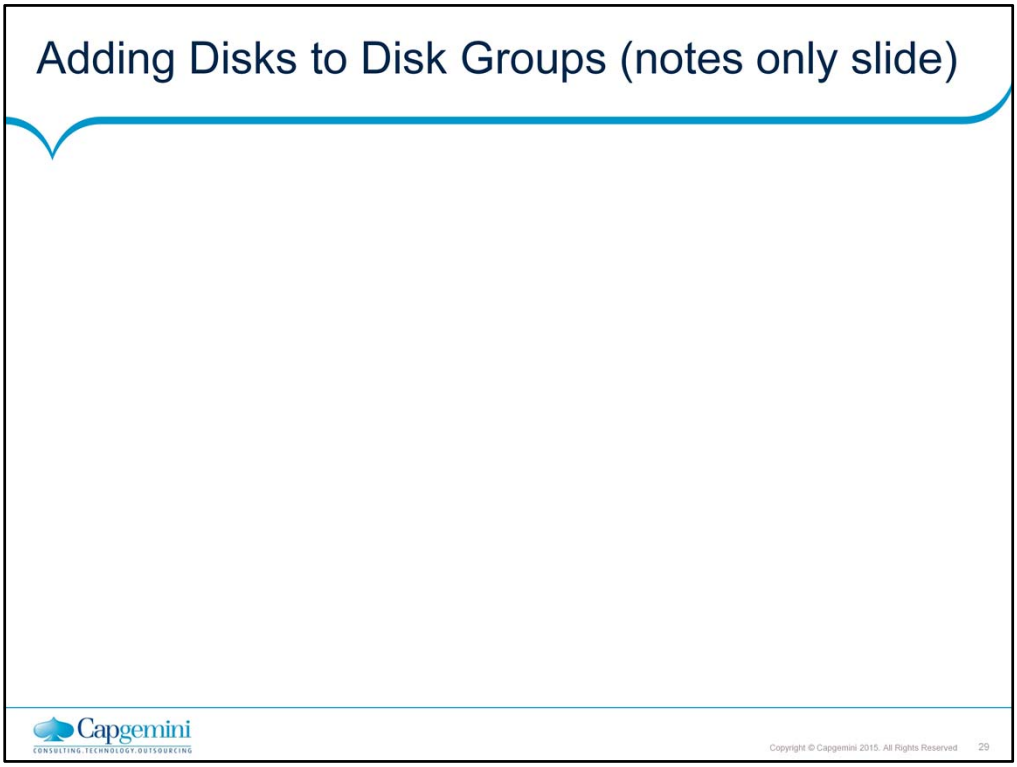

Adding Disks to Disk Groups

This example shows how to add disks to a disk group. You execute an ALTER DISKGROUP ADD DISK command to add the disks. The first statement adds four new disks to the DGROUPA disk group.

The second statement demonstrates the interactions of discovery strings. Consider the following configuration:

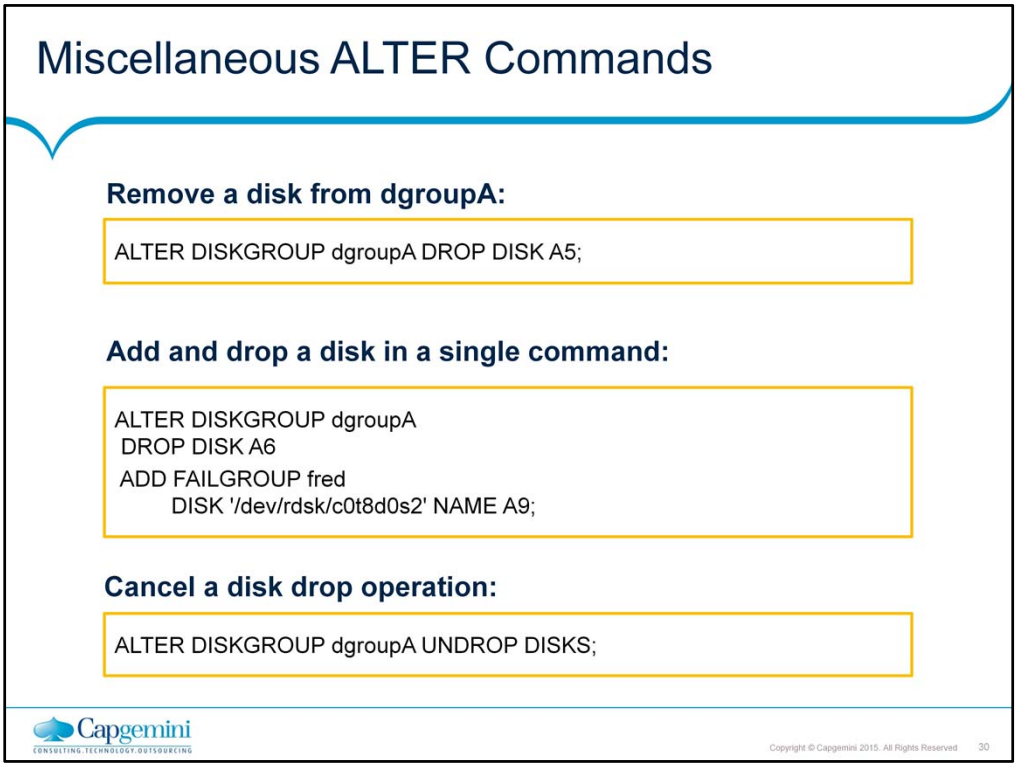
/devices/A1 is a member of disk group DGROUPA.
/devices/A2 is a member of disk group DGROUPA.
/devices/A3 is a member of disk group DGROUPA.
/devices/A4 is a candidate disk.

The second command adds A4 to the DGROUPA disk group. It ignores the other disks, even though they match the discovery string, because they are already part of the DGROUPA disk group. As shown by the diagram, when you add a disk to a disk group, the ASM instance ensures that the disk is addressable and usable. The disk is then formatted and rebalanced. The rebalance process is time consuming because it moves AUs from every file onto the new disk.




Adding Disks to Disk Groups (continued)

Note: Rebalancing does not block any database operations. The main impact that a rebalance process has is on the I/O load on the system. The higher the power of the rebalance, the more I/O load it puts on the system. Thus less I/O bandwidth is available for database I/Os.




Miscellaneous ALTER Commands

The first statement in the slide shows how to remove one of the disks from the DGROUPA disk group. The second statement shows how you can add and drop a disk in a single command. The big advantage in this case is that rebalancing is not started until the command completes. The third statement shows how to cancel a disk drop operation. The UNDROP command operates only on pending drops of disks; it has no effect on drops that have completed.

The following statement rebalances the DGROUPB disk group, if necessary:

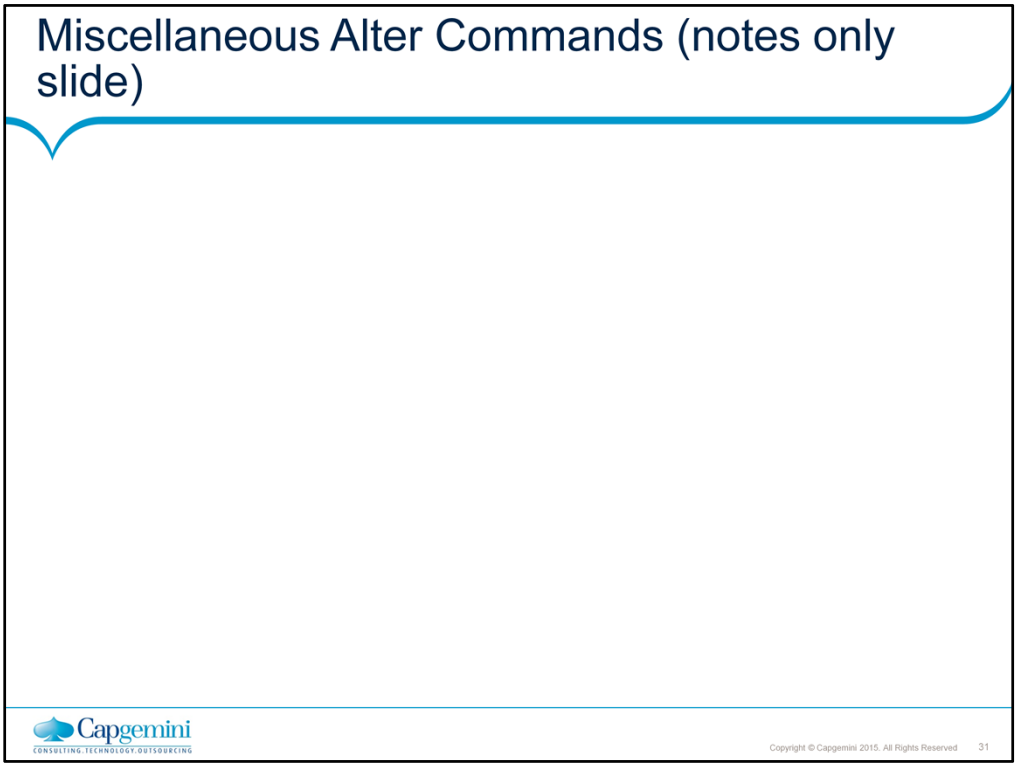
```
ALTER DISKGROUP dgroupB REBALANCE
POWER 5;
```

This command is generally not necessary because it is automatically done as disks are added, dropped, or resized. However, it is useful if you want to use the POWER clause to override the default speed defined by the initialization parameter ASM_POWER_LIMIT. You can change the power level of an ongoing rebalance operation by reentering the command with a new level. A power level of zero causes rebalancing to halt until the command is either implicitly or explicitly reinvoked.

The following statement dismounts DGROUPA:

```
ALTER DISKGROUP dgroupA DISMOUNT;
```

The MOUNT and DISMOUNT options allow you to make one or more disk groups available or unavailable to the database instances.



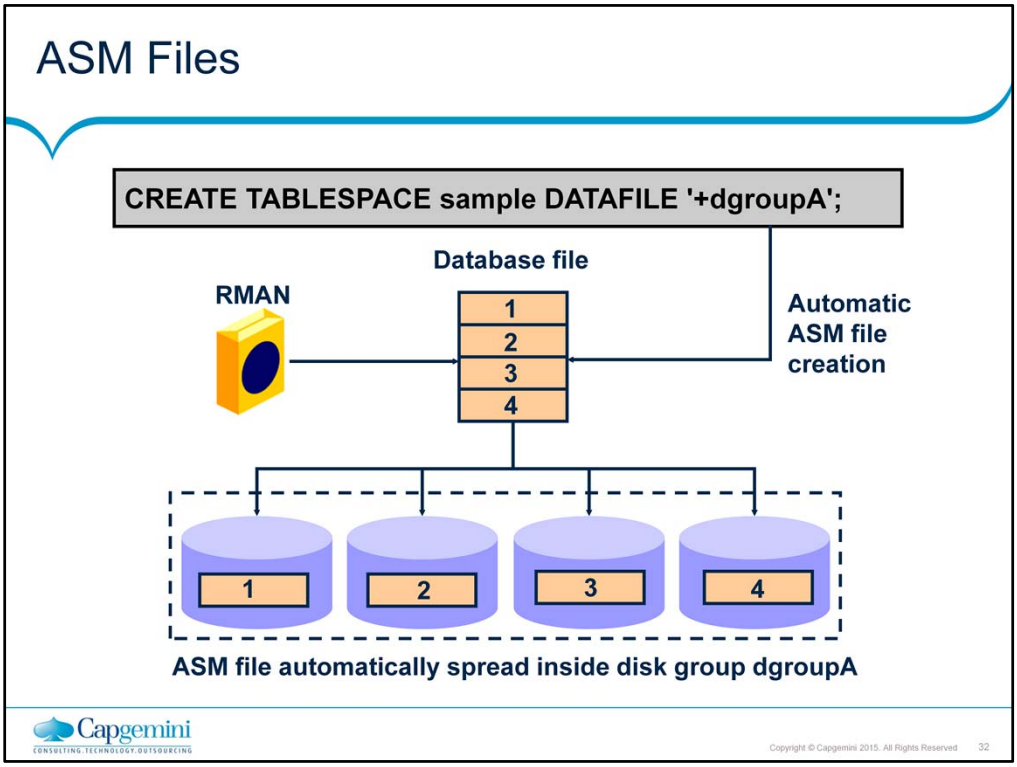

Miscellaneous ALTER Commands (continued)

Use the following statement to verify the internal consistency of disk group metadata and to repair any error found:

```
ALTER DISKGROUP dgroupA CHECK ALL;
```

It is also possible to use the NOREPAIR clause if you just want to be alerted about errors. While the example requests a check across all disks in the disk group, checking can be specified on a file or an individual disk. This command requires that the disk group be mounted. If any error is found, a summary error message is displayed and the details of the detected error are reported in the alert log.

Note: Of these six examples, the first four trigger a disk group rebalancing, and the last two do not.



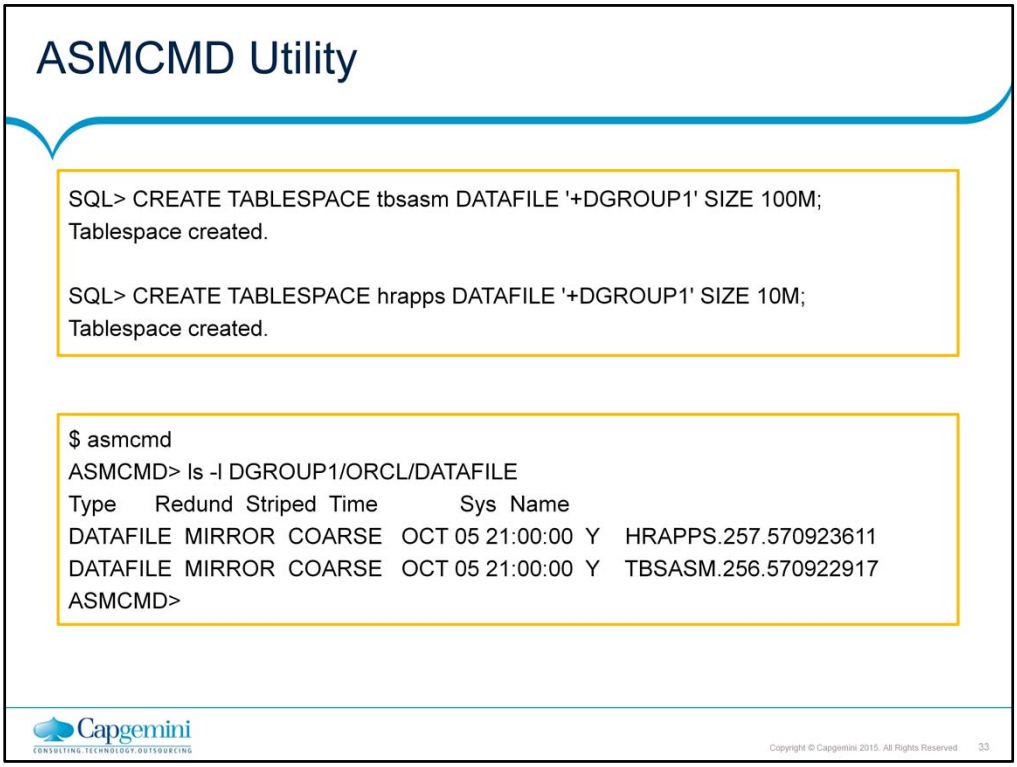

ASM Files

When you specify an ASM disk group as the data file name for a tablespace, ASM files are created in the disk group to provide storage for the tablespace.

When an ASM file is created, certain file attributes are permanently set. Among these are its protection policy, and its striping policy. ASM files are Oracle Managed Files. Any file that is created by ASM is automatically deleted when it is no longer needed.

With ASM, file operations are specified in terms of database objects. Administration of databases never requires knowing the name of a file, though the name of the file is exposed through some data dictionary views, or the ALTER DATABASE BACKUP CONTROLFILE TO TRACE command. Because each file in a disk group is physically spread across all disks in the disk group, a backup of a single disk is not useful. Database backups of ASM files must be made with RMAN.

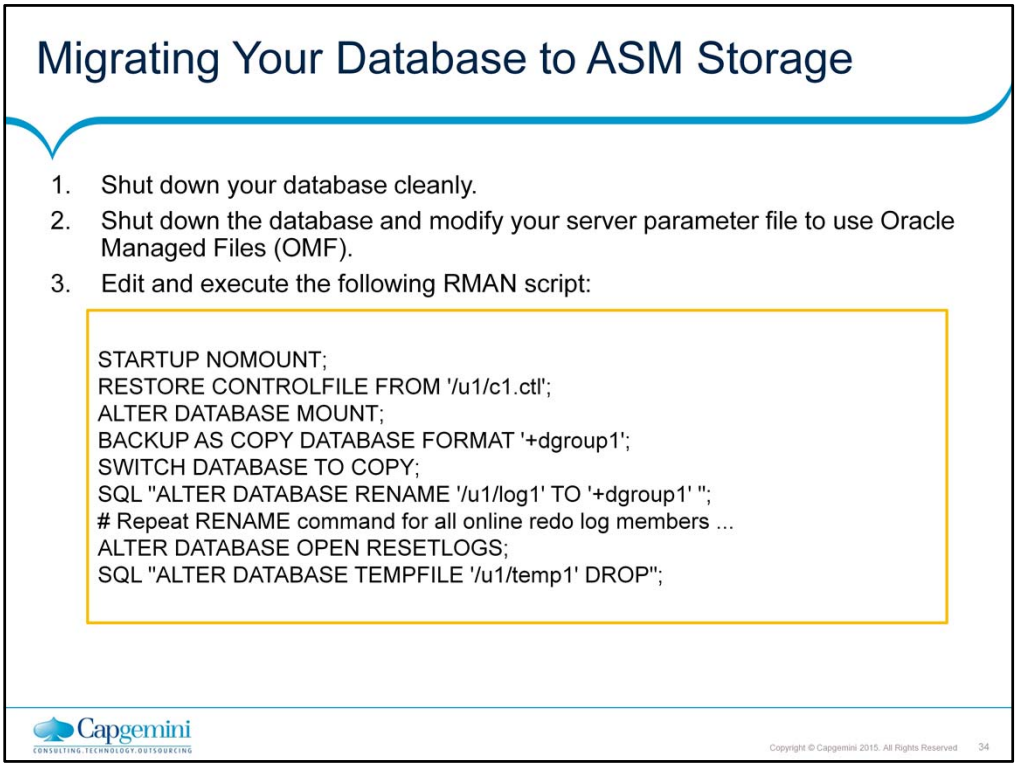
Note: ASM does not manage binaries, alert logs, trace files, or password files.




ASMCMD Utility

ASMCMD is a command-line utility that you can use to easily view and manipulate files and directories within ASM disk groups. It can list the contents of disk groups, perform searches, create and remove directories, and display space utilization, among other things.

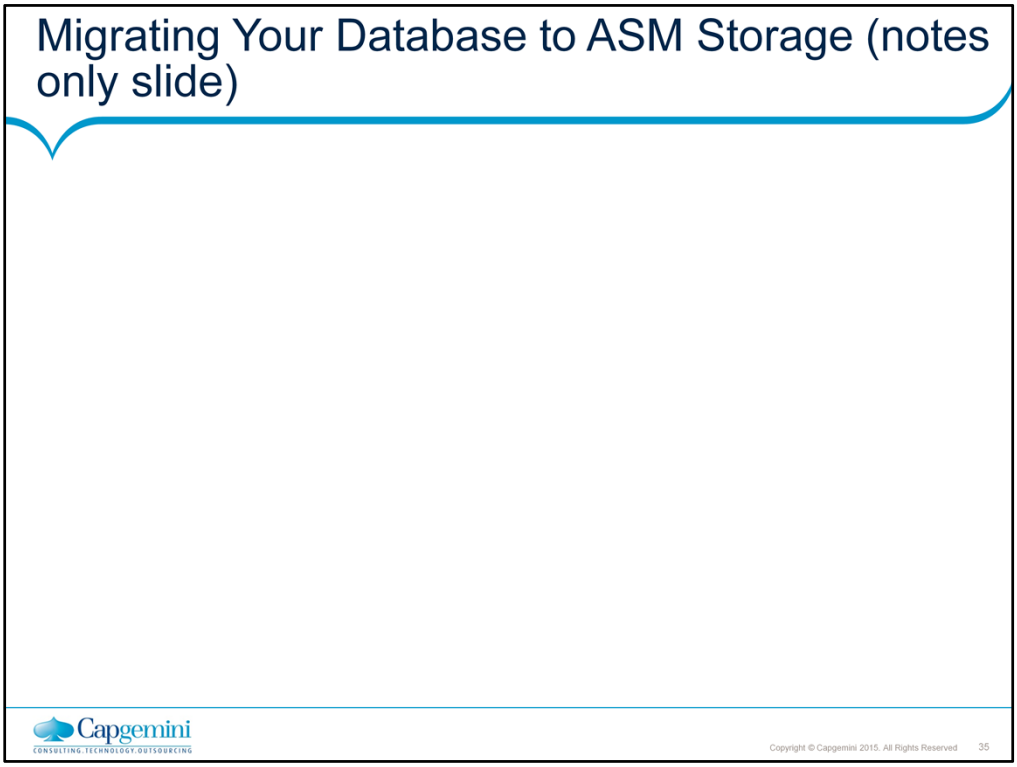
Note: For more information about ASMCMD, see the Oracle Database Utilities documentation.




Migrating Your Database to ASM Storage

Because ASM files cannot be accessed through normal operating system interfaces, RMAN is the only means for copying ASM files. Although files in a tablespace may be 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. You can use the following procedure to relocate your entire database to an ASM disk group: (It is assumed that you are using a server parameter file.)

1. Obtain the file names of the current control files and online redo logs by using V$CONTROLFILE and V$LOGFILE.
2. Shut down the database consistently. Modify the server parameter file of your database as follows:
   - Set the necessary OMF destination parameters to the desired ASM disk group.
   - Remove the CONTROL_FILES parameter.




Migrating Your Database to ASM Storage (continued)

3. Edit and run the RMAN command file, which backs up the database, switches the current data files to the backups, and renames the online redo logs. You can move only tablespaces or data files by using the BACKUP AS COPY command.
4. Delete the old database files.

Note: If you create an OMF control file, and if there is a server parameter file, then a CONTROL_FILES initialization parameter entry is created in the server parameter file.

See the Oracle Database Backup and Recovery Advanced User's Guide for details about how to migrate a database to ASM.

# Summary

- In this lesson, you should have learned how to:
  - Describe the concepts of Automatic Storage Management (ASM)
  - Set up initialization parameter files for ASM and database instances
  - Execute SQL commands with ASM file names
  - Start up and shut down ASM instances
  - Administer ASM disk groups
  - Use RMAN to migrate your database to ASM


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 36